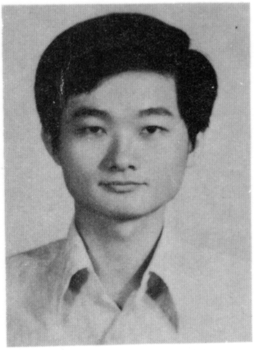
Apple Ⅱ程式集—Game與應用

施威銘　編著



全華科技圖書股份有限公司　印行

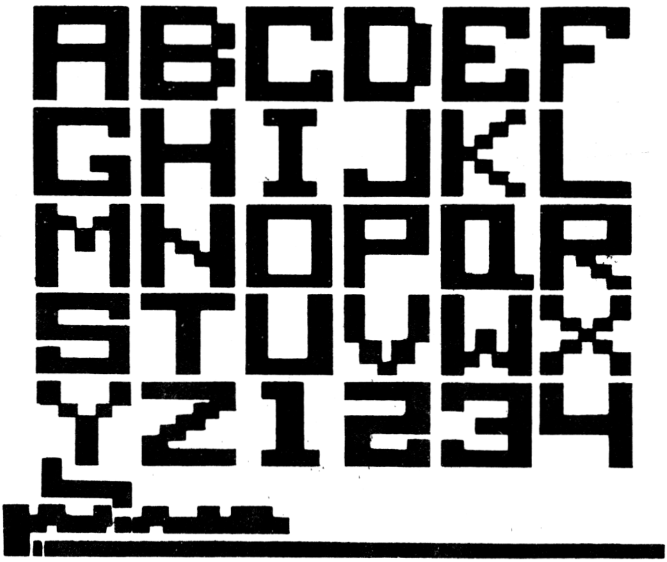


作者小傳

施威銘先生 台灣省彰化鹿港人，民國42年生。台灣大學物理研究所碩士，民國65年並榮獲教育部青年創作發明甲等獎。現任教於健行工專，波前電腦公司、家威影音有限公司技術顧問。著有 Apple Ⅱ 程式集—GAME與應用，Apple Ⅱ 徹底研究(一) Apple Ⅱ 徹底研究(二)等著作多種。

Apple Ⅱ程式集-Game與應用

施威銘　編著



全華科技圖書股份有限公司　印行

感謝您

感謝您選購全華圖書！

希望本書能滿足您求知的慾望！

圖書之可貴 在其量也在其質

量指圖書內容充實、質指資料新穎夠水
準，我們就是本著這個原則，竭心
盡力 地爲國家科學中文化努力
，貢獻給您這一本全是精
華的全華圖書。

序　言

近年來，微電腦技術之進步，其普及化已廣至個人及家用的層面了。不但機種、功能繁多精巧，而且其應用上更是深具彈性而變化多端。

由於這樣廣泛而多樣的使用，使得微型電腦在其本質上大異於傳統。譬如其即時（REAL　TIME）功能，方便於做控制的工作等特性，均爲大型電腦在彈性上、價格上所不及的。可是要將這些特性完全的發揮却是有賴於對每部電腦本身的特性甚至硬體結構加以瞭解才可達成，這同時也是其個人化所在的原因。因此，必須要選定特定的機型，加以深入的探究，，才有辦法發展出微電腦多彩多姿的應用來。

本書卽稟此目的，以目前廣泛使用的APPLE Ⅱ電腦爲例，收集一系列程式，希望借着廣泛的覽視各樣技巧而能研習其長處、特點，以爲參考之用，也可補傳統學習之不足。全部程式均以APPLESOFT BASIC寫成，至於它種高階語言或機械語言的程式，希望能收集另成一書，以適合不同需要的人。

本書之編排分成兩個部份。第一部份是針對APPLE Ⅱ一些常用的特

徵加以提要，並舉例說明之。在這一部份，首先將APPLE的記憶位址結構做扼要說明。另外，一些常用以PEEK、POKE的位址及CALL的副程式均做提要或舉例。高解像畫面（ HIRES ）是APPLE很突出的特色，因此也用詳細的例子來說明。通常高解像畫面圖形與字幕是分開的，在此也舉了一例說明如何在HIRES將圖形與文字混合起來。

在第二部份，共收集了十五個例子，因其個人及家用之傾向，因而多偏於遊戲（GAME）、教育、及私人應用上，一方面是爲維持研習上的興趣，一方面也因其多樣化而明瞭各樣的技巧。在編排上，各程式是相互穿揷而不依難易次序。每個程式均作提要描述，列出程式及說明，但是少數爲免煩長的重複而只列出程式。

只要對APPLE電腦有基本上概念的人，均可很容易的研習、使用本書，如果是對APPLE不熟悉的話，可參考APPLE的使用手册。最後，野人獻曝，如果本書能給需要的人提供一點參考的話，就是最大的願望了。

施威銘

一九八二年七月一日

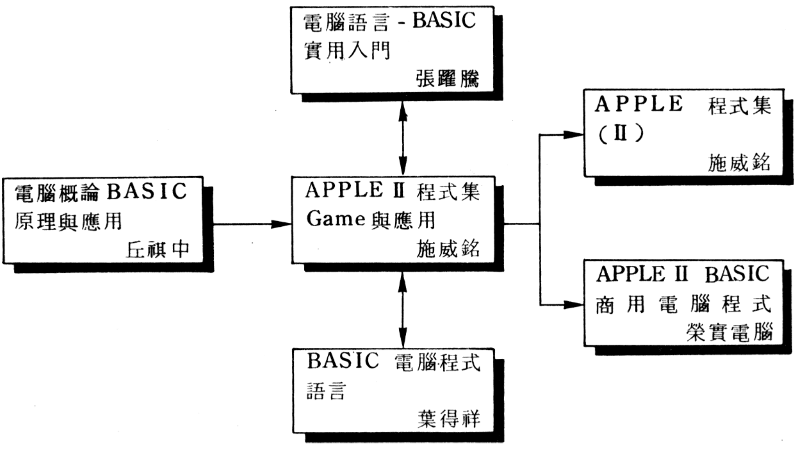
編輯部序

「系統編輯」是我們的編輯方針，我們將提供給您的，絕不只是一本書，而是關於這門學問的所有知識，它們由淺入深，循序漸進。

現在，我們將這本「APPLE Ⅱ程式集－GAME與應用」呈獻給您。近年來微電腦技術之進步，其普及化已經走入了個人及家庭的層面了。家用電腦在國內廣受採用的是APPLE Ⅱ，本書即採用APPLE Ⅱ爲例，全書分成兩部份，第一部份將APPLE Ⅱ一些常用的特性加以提要，並舉例說明之。第二部份共收集了十五個例子，有遊戲、教育、及私人應用程式，從研習上的趣味，學習各種程式設計的技巧。本書實爲練習程式設計最佳參考書籍。

同時爲了使您能有系統且循序漸近研習微電腦方面的叢書，我們以流程圖方式，列出各有關圖書的閱讀順序，以減少您研習此門學問的摸索時間，並能對這門學問有完整的知識。若您在這方面有任何問題，歡迎來函連繫，我們將竭誠爲您服務。

流程圖：


電腦語言 - BASIC
實用入門
張躍騰
電腦概論BASIC
原理與應用
丘祺中
APPLE Ⅱ 程式集
Game與應用
施威銘
APPLE 程式集
（Ⅱ）
施威銘
APPLE Ⅱ BASIC
商用電腦程式
榮實電腦
BASIC 電腦程式
語言
葉得祥

目 錄

第一部份　APPLE Ⅱ程式編寫提要…………………… 1

1-1　APPLE Ⅱ記憶區的結構…………………………… 2

1-2　常用的PEEK及POKE位址……………………………… 5

1-3　常用的CALL機械語言副程式………………………… 11

1-4　在高解像畫面上造出圖案…………………………… 18

1-5　在高解像畫面上作出文數字與圖形單元………… 34

第二部份　程式集…………………………………………… 51

程式一　不會停止的圖案………………………………… 52

程式二　美麗的彩色文字………………………………… 56

程式三　潛水艇（GAME）………………………………… 71

程式四　賽馬（GAME）…………………………………… 90

程式五　顏色的密碼（GAME）…………………………… 97

程式六　跳蛙（GAME）……………………………………109

程式七　分解因數……………………………………………114

程式八　學習英語字母……………………………………………………119
程式九　行星軌道……………………………………………………………123
程式十　代數（解多元一次方程式）………………………………………129
程式十一　電話號碼簿………………………………………………………154
程式十二　吃角子老虎（GAME）…………………………………………182
程式十三　APPLE輪盤（GAME）…………………………………………199
程式十四　APPLE時鐘………………………………………………………209
程式十五　太空防衛戰（GAME）…………………………………………218
附錄〔A〕　十六進位　二進位與十六進位　十進位轉換表…………235
附錄〔B〕　ASCII 碼…………………………………………………………239
附錄〔C〕　參考資料…………………………………………………………241

第一部份

APPLE II 程式編寫提要

在個人用電腦（PERSONAL COMPUTER）領域裡，由於其個人化及家庭化的導向，在小小一部微電腦裡頭，其結構之精緻巧妙實在是令人嘆爲觀止，而在各廠家產品競爭之下，產品功能更是花樣繁多、互爭長短。就以電腦的語言來說，每個廠家爲了擴大其語言能力，單就最普遍使用的BASIC語言，便有不下數十種之多，而這些所謂的“擴展BASIC（EXTENDED BASIC）”或“小型BASIC（TINY BASIC）”語言更是與各自機器本身的系統及硬體結構有着緊密的關聯。因此，對於電腦的使用者來說，不但各種電腦之間缺乏程式的互換性，而且若要在程式編寫上將電腦本身的能力發揮到淋漓盡緻的程度，則非對所使用的電腦本身的系統及內部結構有相當的瞭解不可。

本書第一部份的目的即在於爲了對APPLE電腦做一些概略性的瞭解，以便往後程式編寫之需，而同時也能對初學者在程式之解讀上提供一點幫助。關於進一步的詳細研究，有興趣的讀者可參閱附錄所列之參考書目及資料（R1～R4）。

1-1 APPLE Ⅱ記憶區的結構

由於 APPLE Ⅱ是使用6502微處理晶片做為CPU（即中央處理單元），它的輸出入埠（I/O PORT）所佔用的位址也屬於記憶區的一部份(R5)，因此 APPLE 的記憶區分成三個部門：即由讀／寫記憶器（RAM）、唯讀記憶器(ROM)以及輸出入埠（I/O）所構成。從位址0至位址49151（$0000至$BFFF）* 之記憶區是劃分歸讀寫記憶區RAM來使用的，共有48K位元組（1024個位元組稱為1K位元組），而由第49152位址至53247位址（$C000至$CFFF）共有4K位元組是供給週邊設備及輸出入埠I/O來使用。而最後的12K（總共64K）由位址53248至位址65535（$D000至$FFFF）是由系統監督程式(MONITOR)及BASIC解譯器（INTERPRETER）的ROM所使用的，請參照圖1-1。

使用者由鍵盤鍵入(KEY IN)的程式是存在於48K的RAM記憶區內，當程式執行時這些程式再經由監督程式及解譯器來處理、運算、輸出或輸入等。可是48K的RAM記憶區也不是全部可任意使用的，其中最初的1K記憶區稱為K0（因為通常計數是由0算起的，所以稱K0不稱K1），分別由監督程式及解譯器使用了第0頁區**，程式堆疊使用了第1頁

---

* "$"符號代表十六進位制的數字，有關各種進位制之間的變換關係，請參考附錄〔A〕。

** 十六進制數最低的兩個數位由00算至FF時共佔用了256個位址，這256個位址就稱為一個頁區，同時又利用其前導數字做為編號，譬如$0300～$03FF就稱為第3頁區，$0000～$00FF稱第0頁區。

區，鍵盤輸入緩衝區使用了第 2 頁區，以及大部分的第 3 頁區保留給使用者編寫機械語言使用 (R1)，請參照圖1-2 。接下來是由 $0400 至 $07FF 所構成 1K 位元組的文字幕（ TEXT MODE ）及低解像畫面所共用的螢幕畫面記憶區（ VIDEO RAM AREA ）(R1)，因此眞正可以寫入 BASIC程式的區域是由 $0801 位址開始的 * 。這其間還有各種不同的

頁數：10進位	16進位	
0	$00	RAM (48K)
1	$01	
2	$02	
⋮	⋮	
190	$BE	
191	$BF	
192	$C0	I/O(2K)
193	$C1	
⋮	⋮	
198	$C6	
199	$C7	
200	$C8	I/O ROM(2K)
201	$C9	
⋮	⋮	
206	$CE	
207	$CF	
208	$D0	ROM (12K)
209	$D1	
⋮	⋮	
254	$FE	
255	$FF	

(a)系統記憶圖

頁數：10進位	16進位	用途	
0	$00	系統程式用記憶	
1	$01	系統堆疊(STACK)	
2	$02	輸入暫存區	
3	$03	監督器用記憶	
4	$04	工作及低解析度圖形首頁記憶	
5	$05		
6	$06		
7	$07		
8	$08	工作及低解析度圖形第二頁記憶	可自由使用的記憶頁區
9	$09		
10	$0A		
11	$0B		
12	$0C		
至			
31	$1F		
32	$20	高解析度圖形首頁記憶	
至			
63	$3F		
64	$40	高解析度圖形次頁記憶	
至			
95	$5F		
96	$60		
至			
191	$BF		

(b) RAM結構及其用途

圖 1-1 記憶結構（MEMORY ORGANIZATION）

* $0800 是做爲前導位元組（LEADING BYTE）不可任意使用。

畫面頁區所使用的區間，請參考圖1-1或APPLE手册，當程式使用到時再予詳述。

位　址	十六進制	十進制
Page 0 （第 0 頁區）	$0000 ⋮ ↓ $00FF	0 ⋮ ↓ 255
Page 1 （第 1 頁區）	$0100 ⋮ ↓	256 ⋮ ↓
Page 2 （第 2 頁區）	$0200 ⋮ ↓	512 ⋮ ↓
Page 3 （第 3 頁區）	$0300 ⋮ ↓	768 ⋮ ↓

圖 1-2　記憶空間最初 1K 的位址（K0）

1-2 常用的PEEK及POKE位址

PEEK及POKE是兩個可以直接由BASIC指令將資料（指記憶位址的內含值而不一定是一般高階語言的資料）由記憶內取出或存入的兩個指令（詳見APPLE使用手册），使用者可因此而不必經由呼叫監督程式（CALL －151）而能對記憶單元內的程式資料加以指定或修改。它們的格式是：

POKE I，J

其中I是由0至65535的整個記憶區內的任意一個位址（但注意，由ROM所佔用的位址是不能POKE只能PEEK的），J是由0至255（$00至$FF）的資料（可以是一個機械語言的指令碼，或是位址、數據等）。這個指令表示將J存入I位址內。

X＝PEEK（I）

其中I是由0至65535之間的一個位址，這個敍述表示將I位址的內含值取出並指定為X，亦即X現在的值就和I這個位址的內含值相等，X是0至255的範圍裡。

在APPLE電腦裡，記憶位址大於32767的均可用負號來表示較為方便，譬如65385這個位址寫成－151是一樣的（即將65385這個位址由記憶區倒過來算是第151個位址，65536－65385＝151）。

有些位址在使用上是當成開關來用。一種是像電鈴開關一般，只要用PEEK或POKE來觸動即可（ACTUATE），而不必管其內含值為何，譬如喇叭的發聲位址即是*。另外有一種是像搖頭式的撥動開關一般（見圖1-3），當以PEEK或POKE來觸動某位址，開關就轉換至該位址所控制的狀態。

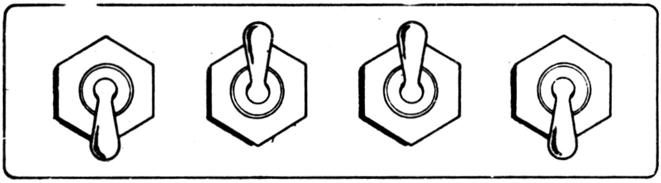


圖 1-3　圖像與字幕的軟體開關位址

字幕窗的控制位址(TEXT WINDOW)**

▶ 32 字幕窗左端邊界

在 32 這個位址所存的數是用來指定文字幕顯示時最左端的位置。如果以 POKE 來改變本位址的內含值，會使得整個文字幕向左或向右移動，但是並不會改變字幕的寬度。如果內含值超過 39 或與字幕寬度的和超

---

* 但是，請嘗試下列程式，是否與內含值無關呢？為何？

```
10   FOR  I=0  TO  255
20   POKE -16336 , I
30   NEXT  I
40   GOTO  10
```

** APPLE電腦可設定螢幕的一部份做為文字顯示用，因此從鍵盤或程式將文字送到螢幕時，只出現在設定的窗口內，因此可用以設定螢幕格式。

過 40 ，則會使字幕超出螢幕範圍，可能會使程式本身或其他重要資料破壞。

▶ 33 字幕窗寬度

此位址的內含值用以設定字幕的寬度。其內含值的範圍是由 1 至 40 之間，改變此值即改變字幕窗右端的位置（左端位置已由 32 位址來設定）。如果將 0 POKE 進入第 33 位址內，BASIC 的解譯器將會被破壞而不能繼續執行工作。如果 33 與 32 兩位址的內含值超過 40 ，可能會使程式本身或其他重要資料受到破壞。

▶ 34 字幕頂端邊界

此位址的內含值用以設定文字幕在螢幕上的頂端位置。此內含值必須在 0 與 23 之間，0 表示最頂端的一列。如果將超過 23 之數值 POKE 進入第 34 位址內，可能會使程式本身或其他重要資料受到破壞。

▶ 35 字幕底端邊界

此位址的內含值用以設定文字幕在螢幕上的底端位置。此內含值必須在 0 與 23 之間，0 表示最頂端的一列。如果將超過 23 之數值 POKE 進入第 35 位址內，可能會使程式本身或其他重要資料受到破壞。

游標(CURSOR)的控制位址

▶ 36 游標的水平位置

設定游標的水平位置，亦即指定下一個所欲顯示出來的字在螢幕上的水平位置。其內含值在 0 與 39 之間，代表游標與字幕左端位置的距離，注意不可使第 32 位址與 36 位址的內含值和超過 39 。

▶ 37 游標的垂直位置

設定游標的垂直位置。其值在 0 與 23 之間，代表游標與螢幕頂端的距離（注意不是字幕頂端），不可將大於23之值 POKE 進入此位址內。

鍵盤位址

▶ －16384 鍵盤讀取位址 $(C000)

由鍵盤送出所按字鍵字碼的位址，亦即讀入鍵盤按鍵資料（字碼）的位址。由於APPLE的字碼比標準 ASCII 多出 128（第 7 位元設定爲 1），因此如果此位址的內含值超過127，則表示有鍵被按了。將此值減去128即是標準的 ASCII 值。

▶ －16368 鍵盤旗標（$C010）

將鍵盤觸發（STROBE）訊號歸零。當APPLE的鍵盤被按了之後，在 －16384 的位址上其第 7 個位元是被設定爲 1 ，必須經由 －16368 位址將 0 POKE 進來，才能將該第 7 位元歸零，以便繼續讀入其他按鍵。

音頻輸出位址

▶ －16352 卡式錄音訊號（$C020）

將欲送往卡式錄音機錄存的訊號經由此位址觸發。

▶ －16336 喇叭發音位址（$C030）

將欲送往喇叭發聲的訊號由此位址來觸發。

圖像開關

這些開關是屬於軟體上的開關，如圖1-3所示，如果 －16304 的位

址被PEEK或POKE來觸動了（即使用該位址），如POKE －16304，0，則螢幕出現一片方格圖案（假設原爲文字幕畫面）。這些位址只受PEEK或POKE的觸動影響而與其眞正數值無關。

▶ －16304 設定螢幕爲圖像畫面（GRAPHIC MODE）（$C050）

利用這個開關來將螢幕設定爲圖像畫面與利用HGR或GR指令不同之處乃在於前者不會將畫面清除。至於所設定的畫面是高解像或低解像畫面，是第一頁區或第二頁區，是全畫面還是混合畫面則由其他開關來決定。

▶ －16303 設定螢幕爲文字幕（$C051）

利用這個開關來將螢幕設定爲文字幕。至於是第一頁區或是第二頁區的文字幕則視其他開關而定。

▶ －16302 設定螢幕爲全畫面（$C052）

利用此開關設定螢幕爲全部圖像畫面。如果當時螢幕是在文字幕狀態，則觸發此開關的結果是不能立即看到的，必須等到－16304的開關被觸發螢幕成爲圖像畫面時才可看出。

▶ －16301 設定螢幕爲混合畫面（$C053）

利用此開關設定圖像畫面的底部有4列的文字幕窗口。如果當時螢幕是在文字幕狀態，則觸發此開關的結果是不能立即看到的，必須等到－16304的開關被觸發，螢幕成爲圖像畫面時才可看出。

▶ －16300 設定爲第一頁區（$C054）

將圖像畫面或文字幕設定爲使用第一頁區的RAM。

▶ －16299 設定爲第二頁區（$C055）

將圖像畫面或文字幕設定爲使用第二頁區的RAM。

▶ －16298 設定螢幕為低解像畫面（$C056）

如果螢幕由 －16304 開關設定為圖像畫面時，此開關則設定螢幕成低解像畫面。這個動作將高解像畫面（ HIRES ）轉換成低解像畫面，但是在記憶位址中，高解像圖形仍然保留。

▶ －16297 設定為高解像畫面（$C057）

如果螢幕由 －16304 開關設定為圖像畫面時，此開關則設定螢幕成高解像畫面 。

因此，依照圖1-4看來，文字幕共有兩種方式（即使用第一頁區或使用第二頁區，但如果是圖形顯示時，則共有八種方式 。

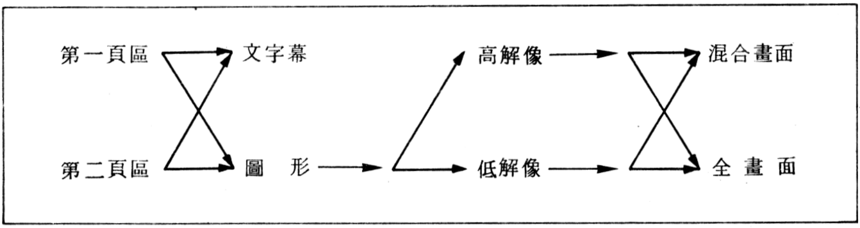


圖 1-4

1-3 常用的CALL機械語言副程式

在RAM的自由區域裡，使用者可以寫上一段機械語言的副程式，然後再由BASIC的主程式來呼叫使用，這個呼叫指令就是CALL，其形式如下：

CALL −384

CALL後面所接的數字代表被呼叫的機械語言副程式的開頭第一個位址，也就是當CALL呼叫之後，程式就進入機械語言的狀態，並且由所呼叫的位址開始執行起，以上面的例子而言，這個位址就是−384（即65536−384＝65152），所以稱−384為該機械語言副程式的"進入點"（ENTRY POINT）。

那麼為什麼要呼叫機械語言副程式呢？因為有些場合，BASIC程式之執行速度太慢，不能適合使用者的要求，這個時候就有必要借助於機械語言的快速執行的優點了。譬如，要令APPLE的喇叭發出一段音樂或是較高音調的聲音，則必須快速振動喇叭的紙盆才行，這項工作用"CONTROL G"或是PEEK(−16336)來做是很困難的，因此必須借助於機械語言程式來完成。

要注意的是，要由CALL來呼叫的副程式必須在最後有一RTS指令（指令碼為$60），如此當副程式執行完畢才能轉回主程式，否則會停止在監督程式裡。

▶ CALL −151

設定目前所執行的語言為機械語言，程式不經過BASIC解譯器而直接交由監督程式處理。因此−151即為監督程式的進入點，鍵入此指令後 RETURN ，螢幕上代表BASIC的游標“] ”被“ * ”游標所取代，表示進入監督程式，此時所能接受的語言為機械語言及監督程式指令*。

▶ CALL −380

設定文字幕為黑底白字的正常字幕。和BASIC的NORMAL指令作用相同。

▶ CALL −384

設定文字幕為反白字幕，這時字是黑色而背景是白色。與BASIC的INVERSE作用相同。

▶ CALL −868

將游標右邊該行的字清除。

▶ CALL −936

將文字幕窗內的所有文字清除，同時將游標移到窗口的最左上端位置。與BASIC的HOME指令以及ESC@的作用相同。

▶ CALL −958

將游標所有該列右邊以及以下各列的字從文字窗內清除，但不更動游標位置。

▶ CALL −1994

---

* 有關機械語言及監督程式指令請參照R1，R5。

將低解像畫面第一頁區的上邊 40 行清除爲黑色，如果是在文字幕時，則會將上邊 20 行以反白的@符號填滿，這個副程式對第二頁區或高解像畫面沒有作用。

▶ CALL －1998

與CALL －1994 相同，但是將全部畫面清除，如果是在文字幕時，則以反白的@符號填滿全部螢幕，這個副程式對第二頁區或高解像畫面沒有作用。

▶ CALL 62450

將高解像畫面清除爲黑色底色。

▶ CALL 62454

依照最後一次所畫的顏色，將整個高解像畫面設定以該顏色爲底色。譬如：

```
10   HGR:  HCOLOR=2
15   HPLOT  0,0
20   CALL  62454
```

就可將整個畫面設定爲藍色的底色。

▶ CALL 768

768這個位址換成十六進制就是＄0300 的第三頁區的第一個位址，第三頁區除＄03F0至＄03FF 外均可由程式設計者自由運用，通常若程式不長的話可在這240 位元組（BYTE）的區間內寫下所欲使用的副程式。

【例 1 】將一個音頻產生程式寫入第三頁區（PAGE 3）

解：首先CALL －151，將機械碼存入第三頁區

```
0300:  AD  30  C0  88  D0  05  CE  16  03  F0
```

09 CA D0 F5 AE 15 03 4C 00 03 [RETURN]

然後鍵入0300L [RETURN] 將會看到：

```
0300－  AD 30 C0   LDA  $C030
0303－  88         DEY
0304－  D0 05      BNE  $030B
0306－  CE 16 03   DEC  $0316
0309－  F0 09      BEQ  $0314
030B－  CA         DEX
030C－  D0 F5      BNE  $0303
030E－  AE 15 03   LDX  $0315
0311－  4C 00 03   JMP  $0300
0314－  60         RTS
0315－  ××         ???
0316－  ××         ???
```

這是曾經載於舊版APPLE手册上的音頻產生程式，其方式是利用觸動喇叭發聲位址$C030（即－16336），記得在1-2我們曾經提到用PEEK（－16336）來觸動喇叭發聲，但是由於使用的是BASIC語言，因此觸動速度太慢，頻率太低，以至於所發出來的聲音太過低沈。現在在$0300位址上用了LDA $C030來觸動喇叭，因爲用的是機械語言，自然速度很快而頻率很高，甚至是高過耳朵能聽聞的範圍，因此必須利用由接下來$0303至$030D的時間延遲程式來控制喇叭振動頻率在適當的範圍，而這個頻率是由$030E的LDX $0315來設定的，如果$0315位址的內含值較大，則因$030B及$030C的DEX與BNE的緣故使得延時廻路的時間較長，因而$0311的JMP至$0300來觸動喇叭的時間也較長，所以喇叭每秒鐘振動次數也就較少了，發出來聲音就

低。反之，若 $0315 位址的內含值較小的話，聲音就高。至於喇叭被觸動發聲要到什麼時候停止呢？在迴路中我們加入了 $0306 至 $030A 的三個指令來檢知 $0316 的內含值，每次迴路走過一次，$0316 的值就減一，當 $0316 的值爲零時就跳出迴路至 $0314 的 RTS 而回到主程式。因此 $0316 的值越大時發聲的時間就長，反之則短。

總之 $0315 位址的內含值是決定音的高低，而 $0316 位址的內含值是決定該音的長短。此時只要在BASIC的主程式裡用 POKE 789，P及POKE 790，D 然後 CALL 768 就能令喇叭發出由P所指定的音高（PITCH）及D所指定的音長（DURATION）的聲音了。

可是，當我們用 SAVE 指令將程式錄到磁帶之後，將機器關掉再用LOAD將程式讀回來之後，發現聲音沒有了，而在第三頁區的程式也不見了。這是因爲SAVE及LOAD指令是BASIC指令，只用來存取BASIC程式用的，可是BASIC程式是由 $0801 開始的，自然第三頁區的副程式就消失了。爲了解決這個問題，通常在程式不太長的情況下（所以才寫在第三頁區），是在BASIC主程式內將聲音副程式用 POKE 指令存入第三頁區的，如此聲音副程式就可存到磁帶了。

【例2】將例1的機械語言副程式以BASIC語言存入第三頁區，並呼叫使用。

```
0     REM ** BASIC POKES
10    GOSUB 1000
20    FOR I=0 TO 255 STEP 10
30    POKE 789,I: CALL 768
40    NEXT I
50    POKE 789,100
60    FOR J=0 TO 255 STEP 10
70    POKE 790,J: CALL 768
80    NEXT J
```

```
90      END
1000    FOR K=0 TO 20: READ M: POKE
        768+I,M: NEXT K: RETURN
1100    DATA 173,48,192,136,208,5,206,22,3,
             240,9,202,208,245,174,21,3,76,
             0.3,96
```

在這個例子中，必須要先將機械語言的十六進位碼化成十進位數*，再放入至DATA內以便讀取爲POKE用。寫程式的途徑很多，下面是發聲程式的另一個例子**，往後在其他場合還會看到更多的程式，但其基本原則應該是相類似的。

【例3】另一個發聲副程式

```
0300—  ××          ???
0301—  ××          ???
0302—  AC 01 03    LDY    $0301
0305—  AE 01 03    LDX    $0301
0308—  A9 04       LDA    #$04
030A—  20 A8 FC    JSR    $FCA8
030D—  AD 30 C0    LDA    $C030
0310—  E8          INX
0311—  D0 FD       BNE    $0310
0313—  88          DEY
0314—  D0 EF       BNE    $0305
0316—  CE 00 03    DEC    $0300
0319—  D0 E7       BNE    $0302
031B—  60          RTS
```

* 請參考附錄A。

** R8。

此程式是由 \$0300 及 \$0301 兩個位址的內含值來定出聲音的長短（D）及高低（P），所以由BASIC來叫用時是CALL 770 而不是768。另外，程式裡用到\$FCA8監督程式的內藏副程式WAIT，這個副程式是將累算器（ACCUMULATOR）的內含值A以½（$26+27A+5A^2$）微秒的時間來做延時。在APPLE的監督程式內部具有許多這種有用的副程式，適當的加以引用對於程式編寫會有很大幫助的。

1-4 在高解像畫面上造出圖案

在一些電視遊樂器上，或是電腦遊戲程式裡頭，通常需要將某些特定的圖形在畫面上產生出來，並加以上、下、左、右的移動，或是加以旋轉甚至放大或縮小。APPLE電腦有一系列的特殊指令，專門用來做出這些效果的，譬如：DRAW、XDRAW、ROT、……等等，在詳加解釋這些指令的用法之前，先來看一般的BASIC程式是如何處理這種問題，然後再進一步來瞭解APPLE的專用指令，會更能幫助對整個過程的瞭解。

以BASIC來產生圖形

先看底下一個例子（例4），這是以高解像指令HPLOT來將(0,0)、(0,159)、(279,159)三點連成一個直角三角形。以這種方式來做圖的話，如果要畫很多三角形，則必然要使用很多次的HPLOT指令了，這會使程式變得又長又麻煩。因此，在例5的程式中，我們將程式以READ、DATA來將例4加以改進，每次READ一組DATA之後，程式即經由第30行的指令加以計算，並且由第35行的HPLOT將圖形畫出。所以只要改變DATA的數值，就可以畫出不同的三角形，如果再加入INPUT的指令，就可以做出能與使用者交互作用（對談式）的程式了。

【例4】 在APPLE的高解像畫面上，使用HPLOT來畫出直角三角形。

解：

```
10  HGR：  HCOLOR＝3
20  HPLOT  0,0  TO  0,159  TO  279,159  TO  0,0
```

見圖1-5。

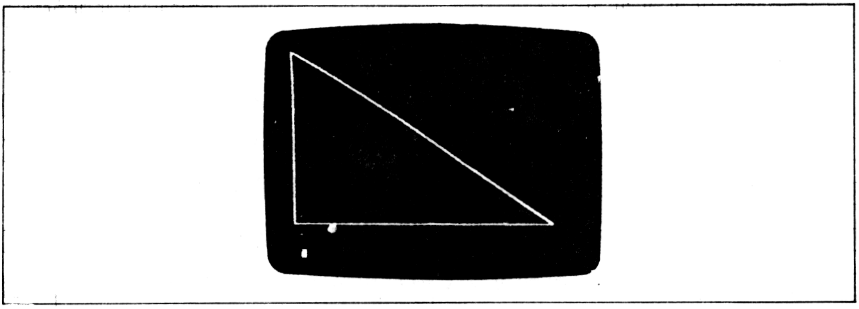


圖 1-5

【例 5】在高解像畫面上畫出其中一個角度爲 30°，45° 及 80° 的三個直角三角形。

解：

```
10 HGR: HCOLOR=3: F=1.212
15 PI=3.14159
20 FOR K=1 TO 3
25 READ I,J,O,T
30 T=90-T: T=T/180*PI: A=O/TAN(T)/F
35 HPLOT I,J TO I-O,J TO O,J-A TO I,J
40 NEXT: END
50 DATA 140,80,70,45,240,120,200,30,50,
        150,30,80
```

見圖1-6。

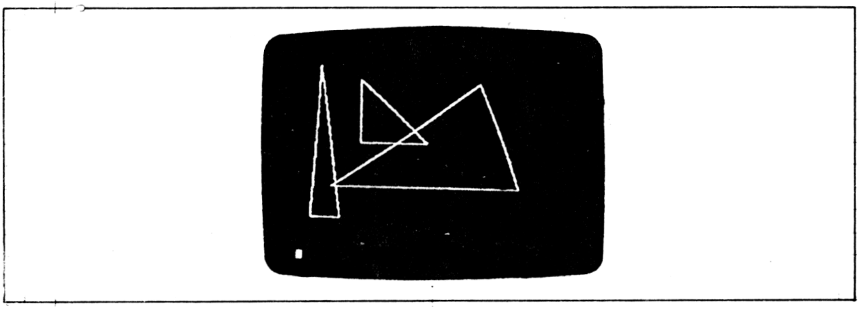


圖 1-6

將圖形加以移動

由以上兩個例子可以看出，要靈活使用圖案的話，可以將圖案上，每一個線段的起點與終點存在DATA裡，然後利用READ指令取出來使用，使用FOR……NEXT指令可重覆READ。如果圖形是封閉的話，則N條線段共有N個點（因爲一條線段的終點，也就是另一條線段的起始點，爲了方便可以規定取其終點卽可），但是在畫圖時是必須回到起始點的（因此起點也是終點）。所以 DATA 所要存的資料是N＋1項。

圖1-7就是利用座標紙將一隻海角的圖案畫出來，並將其線段的終點記錄下來，以便存放到程式的DATA欄裡頭。例7就是將所得DATA畫在畫面上的程式，在程式中利用陣列X(I)及Y(I)先將X座標及Y座標

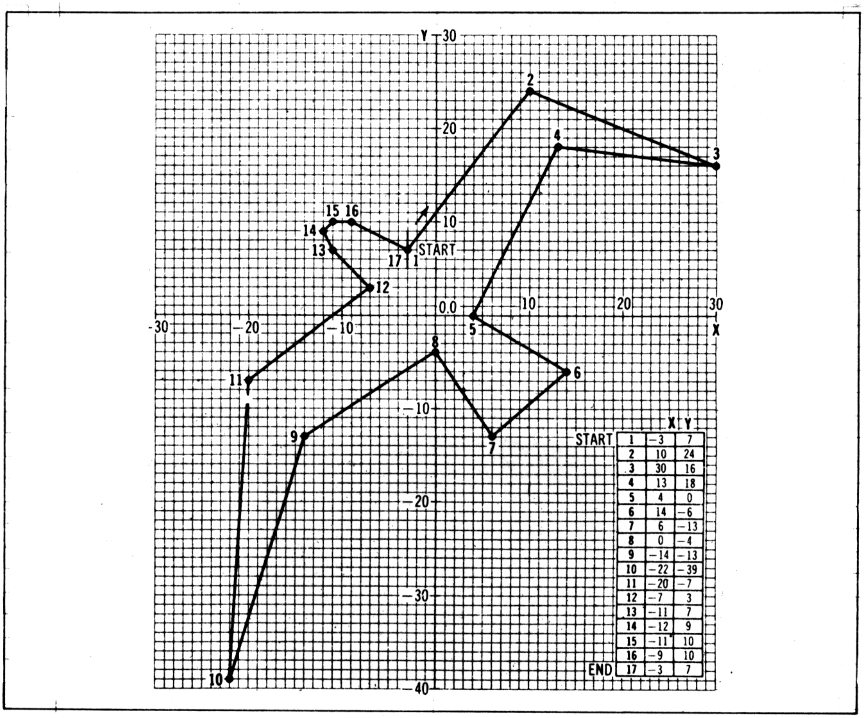


圖 1-7

值讀取，其中Y的值已先縮小1.212，這是因爲在方格紙上的X，Y座標每格的比例是 1：1 而在APPLE電視螢幕上X：Y≃1：1.212 。

例8是將上述的圖形沿著水平方向移動的例子，其中第100行是畫出圖形的副程式，第1000行是讀取X及Y座標值的副程式，第40及60行是將圖形移動的FOR……NEXT迴路，第45行將圖形畫出，第55行的HCOLOR＝0將圖形消除，第50行是延時程式，使畫出來的圖形能稍作暫留。

至此，我們已瞭解了圖形移動的基本原理了，首先要將圖形的座標（即形狀）訂出，然後用各種指令加以運算或處理，以得到吾人所期望的效果。雖然在此我們以BASIC來寫程式，但是在快速處理的機械語言中，其基本原則也是一樣的。

【例6】在方格紙上畫出一個海鳥的圖形，並按順序記錄其每一點的座標值。

注意：其始點座標（－3，7）亦同時爲終點座標。見圖1-7。

【例7】使用圖1-7所得到的（X，Y）座標值來畫出圖形。

解：

```
10   HGR: HCOLOR=3: F=1.212
15   DIM X(16),Y(16)
17   N=16
20   FOR I=0 TO N: READ X(I): NEXT I
25   FOR I=0 TO N: READ Y(I): Y(I)=-Y(I)
     /F: NEXT I
30   HPLOT 140+X(0),80+Y(0)
35   FOR I=1 TO N: HPLOT TO 140+X(I),
     80+Y(I): NEXT I
100  DATA -3,10,30,13,4,14,6,0,-14,-22,
          -20,-7,-11,-12,-11,-9,-3,7,
          24,16,18,0,-6,-13,-4,-13,-39,
          -7,3,7,9,10,10,7
```

見圖1-8。

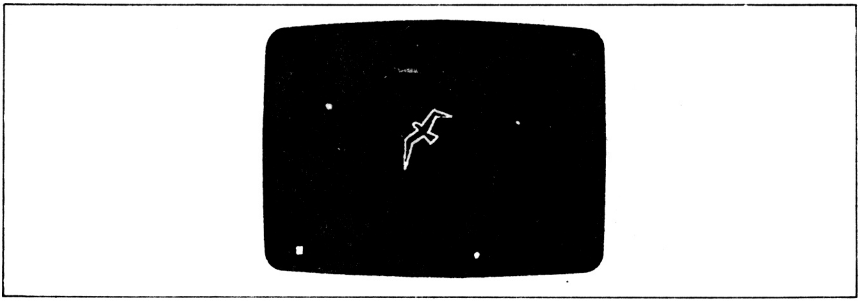


圖 1-8

【例 8】在畫面上做出移動的圖形。

解：
```
10    HGR: HCOLOR=3: F=1.212
15    J=100
20    GOSUB 1000
40    FOR I=240 TO 25 STEP -15
45    HCOLOR=3: GOSUB 100
50    FOR D=1 TO 50: NEXT D
55    HCOLOR=0: GOSUB 100
60    NEXT I
70    GOTO 40
100   HPLOT I+X(0),J+Y(0): FOR K=1 TO
      N: HPLOT TO I+X(K),J+Y(K): NEXT
      K: RETURN
1000  N=16: DIM X(N),Y(N): FOR I=0 TO
      N: READ X(I): NEXT I: FOR I=0 TO
      N: READ Y(I): Y(I)=-Y(I)/F: NEXT
      I: RETURN
1010  DATA -3,10,30,13,4,14,6,0,-14,-22,
           -20,-7,-11,-12,-11,-9,-3,7,
           24,16,18,0,-6,-13,-4,-13,-39,
           -7,3,7,9,10,10,7
```

見圖 1-9。

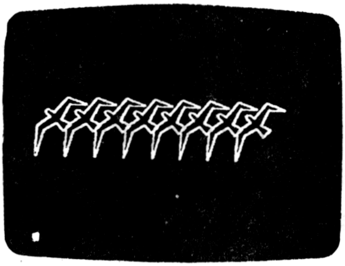


圖 1-9

APPLE圖形表(SHAPE TABLE)指令的使用

上述的幾個例子是相當容易明白的，但是在使用上却有一些不便，譬如當圖形較複雜時執行的速度就變慢了，要使圖形旋轉也是一項困難。另外，在襯有底色背景的圖面上，使用HCOLOR＝0 的方式來消除圖案，會連底色也消除成爲黑色。

因此，在APPLESOFT的指令裡頭，使用了圖形表的方式來解決這些問題，圖形表是使用上、下、左、右四種向量來將圖形描出來，並將描出圖形的這些向量以規定的方式存到記憶單元裡頭。將這些資料存好之後，就可以使用下列一系列的BASIC指令來將原圖形加以運用了。

DRAW I AT X，Y

是將存在記憶單元內的圖形表中的第 I 個（總共有可存255 個）圖形畫在畫面座標爲（X，Y）的位置上。

XDRAW I AT X，Y

使用XDRAW可將先前的圖形消去而又不會消除背景的底色。

ROT＝ K

這個指令會令將由DRAW所畫出的圖形旋轉一個角度（因此必須在DRAW

之前指定ROT)。K可由0至255之間，但只有64種可能轉動的角度。當ROT=0時表示不轉，當SCALE=1時，ROT只承認0,16,32,48等四種角度(分別爲0°,90°,180°,270°)，當ROT=64則轉360°和0°一樣。當SCALE=2時，則可轉8種角度，……依此類推，而不被承認的值則依較小之承認值旋轉，譬如SCALE=1時，ROT=33和ROT=32所轉的角度一樣。

SCALE= J

J可由0到255之間，SCALE=1表示原大小之圖形，SCALE=2表示將繪圖向量放大兩倍(每個向量畫兩次)，……依此類推。
注意：SCALE=0表示圖形放至最大而不是最小。

SHLOAD

是將圖形表之資料由磁帶讀至電腦裡的指令。

將圖形表存至磁帶的方式：

因爲圖形表的資料是存在記憶位址的某一段區間裡頭，因此要叫出監督程式(MONITOR)來，找出圖形表的起始位址$SSSS、終止位址$RRRR以及二者的差$DDDD，然後將差額存入位址$00與$01裡頭：

0: DD DD RETURN
(注意：低位元組先存至$0，高位元組存至$01)

再將表的長短(即差額)以及圖形表本身存入卡帶

0.1W SSSS. RRRRW RETURN

至此，我們尚未指出圖形表製作的方法，以下我們以一個實例來說明就可明白。要造出一個圖形表可依7個步驟來完成，首先將圖形以上、下、左、右四種向量來組成，將向量依次序排列，分解成16進位值，加上

圖形表的起始編號（因爲可能有許多個圖形存在表裡），存入記憶體內，存入卡帶，由卡帶讀入以便使用：

第一步驟　描繪圖形

依照圖 1-10(a)的方式將圖形在方格紙上以上、下、左、右的小箭頭依序組成，圖中的起始位置（START）就是後來使用在ROT指令時之旋轉中心軸。第一個圖只用4個箭頭向量，第二個圖用了12個箭頭向量。

第二步驟　分解向量

如圖 1-10(a)的方式將圖形分解

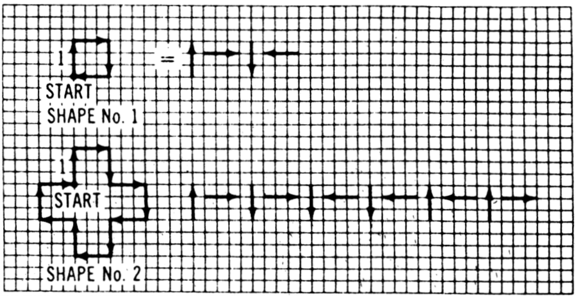


圖 1-10(a)

第三步驟　轉換成16進位數

依照圖 1-10(c)的方式先將每個箭頭向量轉換成代碼，再將這些代碼值轉換成 16 進位數。圖 1-10(c)的塡表方式是依 A→B→C 的方式由每一位元組（BYTE）的最低位元（BIT）塡起，塡完了一個位元組之後再依次塡下一位元組（BYTE）。注意：通常以虛線箭頭向量表示在圖面上不要畫出來，其對應代碼應該是“0DD”。整個轉換過程請參照圖 1-10(b)。

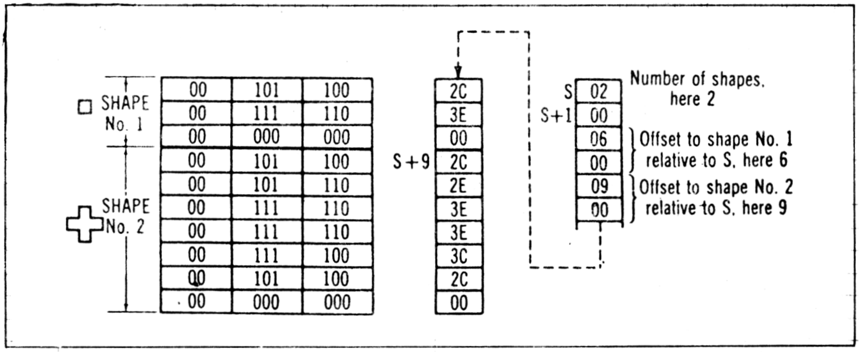


圖 1-10(b)

代碼	C B A 位元號碼： 7 6 5 4 3 2 1 0 指　　定： D D P D D P D D 每組位元對（ bit pair ）DD，指定一個移動的方向，而每個位元 P 指定在移動之前要不要畫個點。如下面所說： 如果DD＝00 向上移　　如果P＝0 不畫 ＝01 向右移　　＝1 畫 ＝10 向下移 ＝11 向左移

圖 1-10(c)

第四步驟　加入圖形編組

因爲圖形表最多可塡入255個不同的圖形，因此必須加以編號才能區分。在圖形表第一位元組（ BYTE ）的值代表圖形的個數，第二位元組要塡 00 ，第三、四二個位址就用來塡第一號圖形的起始位址與圖形表第一位址的差（注意：機械語的高低位址要倒過來塡），其它類推，可參照 APPLE 程式手册。

第五步驟　將圖形表存入記憶體

首先將圖形表的長度存入00及01的位址，在本例中共使用了16個位元組，因此16－1＝15＝$0F（最後位址減去最前位址＝位址差額）。

0： 0F 00 　RETURN

接下來把圖形表存入記憶體，如果表不太長的話，可存在第三頁區內，若太長則存在較高位址的地方，要注意到使用HIMEM來設定BASIC程式範圍，以免和圖形表相衝突。譬如：

8000： 02 00 06 00……2C……2C 00 　RETURN

第六步驟　存入卡帶

依前述存入磁帶方式將圖形表的長度以及表本身存入磁帶。

0.1W 8000. 800FW 　RETURN

第七步驟　由磁帶讀取圖形表

只要將卡帶依一般方式以SHLOAD指令讀取即可。雖然是以監督程式指令來存入磁帶資料，但讀取時可以不必指定起始及終止位址而以SHLOAD很方便的就可讀回資料。

（注意：如果圖形表不是用SHLOAD存入，則必須如下例將起始位址存入$E8，$E9。）

程式範例

將上述的圖形表加以利用可以組成以下的示範程式。程式由第0行跳至第1000行，在第1005行將圖形表的起始位址0300存入$E8，$E9零頁區的位址上，第1015行的FOR……NEXT迴路將第1030行的

DATA（其實就是圖形表，只不過是以十進位的方式表示出來，以便在BASIC中以POKE指令來存入記憶體）以第1020行來讀入並POKE至0300（\$0300＝768₁₀）之後的16個位址內。1200行設定各變數值。

第2000行至2020行印出4種圖形顯示方式的目錄，第2021行將畫面設定爲圖形與字幕混合形式以便在圖形底部印出4行字幕，同時以GET指令等候選擇（1～4），第2045行依選擇（S）而分別跳至100 EXPLODE，200 VORTEX，300 SNOWFLAKE，400 SELECTION 等4種圖案顯示。

在EXPLODE中先將畫面以HGR及 POKE －16302，0 設定爲全畫面顯示，第115行以HCOLOR＝0及CALL 62454將畫面設定爲全黑色背景並跳至第50行副程式設定非黑色顏色，然後以不同的SCALE畫出圖形（2030行指定 SH＝1或2）。

在VORTEX中，置定全畫面之後以GOSUB 50及CALL 62454來指定不同的背景底色，注意第215行L＝NOT L是爲了得到BLACK 1與BLACK 2以得到不同的線條效果（有興趣的讀者請參考R1），另外依L的值也使VORTEX可以是擴展或收縮（220行及230行），第235行的ROT＝I及SCALE＝I就造出了“VORTEX”的效果。

在SNOWFLAKE中，是第345行ROT＝I，及第335行SCALE＝H的雙重迴路，因此和VORTEX不同。

在FALLING中，只有指定SCALE＝I的FOR……NEXT迴路，ROT並沒指定（因此旋轉角度爲0°）。另外，由本例子可發現，當圖形表不是很長而且只要用在一個程式時，可以用POKE指令來存入而不必另外以SHLOAD來由磁帶讀，如此在使用上較爲方便。

程 式：

```
] LIST
0    GOTO 1000
```

```
50    C= INT (RND (1)*6+1): HCOLOR=
      C: IF C=4 THEN HCOLOR= 3
55    RETURN
100   REM EXPLODE
105   HGR: POKE -16302,0: X=140: Y=95
110   FOR M1= 1 TO MAX
115   HCOLOR= 0: HPLOT 0,0: CALL 62454:
      GOSUB 50: ROT= RND (1)*64
120   FOR I=1 TO M STEP M1
125   SCALE= I
130   DRAW SH AT X,Y
135   NEXT I,M1
140   GOTO 0
200   REM VORTEX
205   HGR: POKE -16302,0: X=140: Y=95
210   GOSUB 50: HPLOT 0,0: CALL 62454
215   L= NOT L: HCOLOR= L*4: IF L
      THEN 230
220   FOR I=1 TO M
225   GOTO 235
230   FOR I=M TO 1 STEP -1
235   ROT= I: SCALE= I
240   DRAW SH AT X,Y
245   NEXT I
250   J=J+1: IF J=6 THEN J=0: GOTO 0
255   GOTO 210
300   REM SNOWFLAKE
305   HGR: POKE -16302,0
310   HCOLOR= 0: HPLOT 0,0: CALL 62454
315   L= NOT L: IF L THEN 330
320   FOR H=5 TO 50 STEP 7
```

```
325  GOTO 335
330  FOR H=50 TO 5 STEP -1
335  SCALE= H: GOSUB 50
340  FOR I=1 TO M STEP RND (1)*6+2
345  ROT= I
350  DRAW SH AT X,Y
355  NEXT I,H: FOR D=1 TO 1000: NEXT
     D
360  J=J+1: IF J>4 THEN J=0: GOTO 0
365  GOTO 310
400  REM FALLING
406  HGR: POKE -16302,0
410  HCOLOR= 0: HPLOT 0,0: CALL 62454
415  M1= RND (1)*3+1: X=140: Y=95
420  FOR I=1 TO M STEP M1
425  SCALE= I: GOSUB 50
430  DRAW SH AT X,Y
435  X=X+1: Y=Y+1: IF X>278 OR Y>191
     THEN X=140: Y=95: GOTO 445
440  NEXT I
445  J=J+1: IF J>6 THEN J=0: GOTO 0
450  GOTO 410
1000 REM LOADS THE SHAPE TABLE START
     ADDRESS
1005 POKE 232,0: POKE 233,3
1006 RESTORE
1010 REM LOADS THE ACTUAL TABLE DATA
1015 FOR LOC= 768 TO 783
```

```
1020 READ BYTE: POKE LOC, BYTE
1025 NEXT LOC
1030 DATA 2,0,6,0,9,0,44,62,0,44,46,62,
          62,60,44,0
1200 M=60: M1=1: MAX=15: ROT=0:
     SCALE=5: X=140: Y=95: CNT=25
2000 REM MENU STARTS HERE
2001 HOME: VTAB 21
2005 PRINT "1 EXPLODE 2 VORTEX 3 SNO-
     WFLAKE"
2010 PRINT "4 FALLING"
2020 PRINT "SELECTION ?";
2021 POKE -16301,0: GET S
2025 IF S<1 OR S>4 THEN 2015
2030 K= NOT K: SH= K+1
2035 IF SH= 2 THEN M=60
2040 IF SH= 1 THEN M=80
2045 ON S GOTO 100,200,300,400
```

執行例:

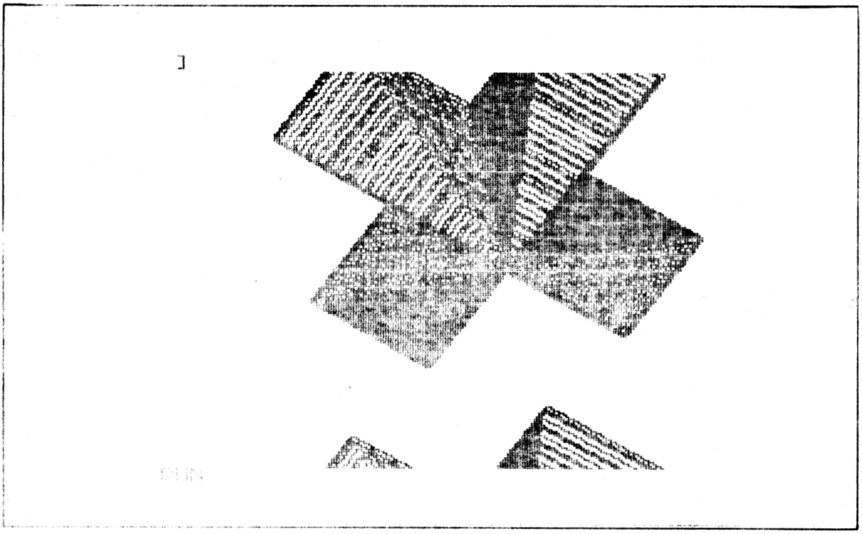


圖 1-11(a)

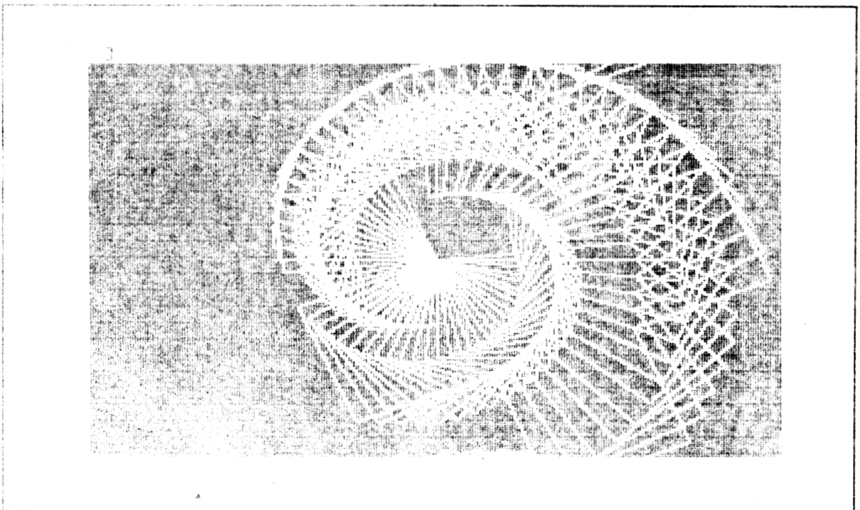


圖 1-11(b)

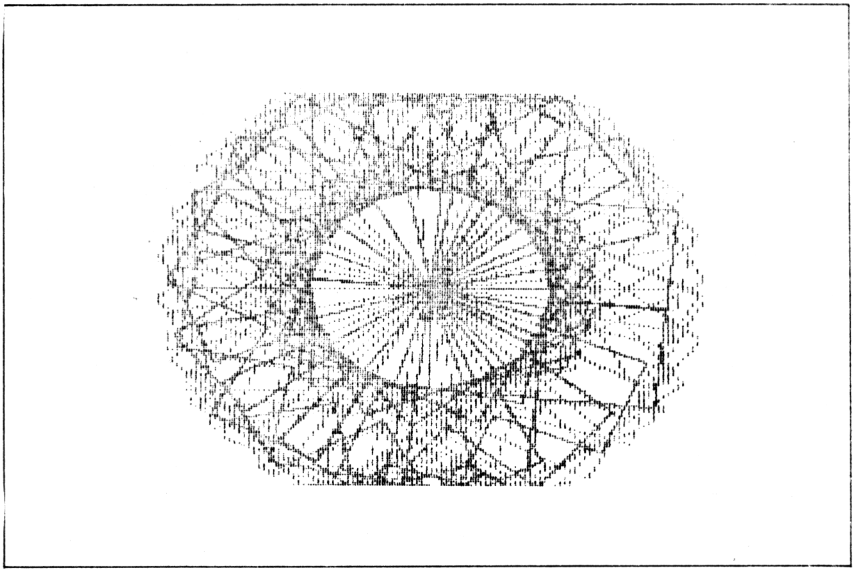


圖 1-11(c)

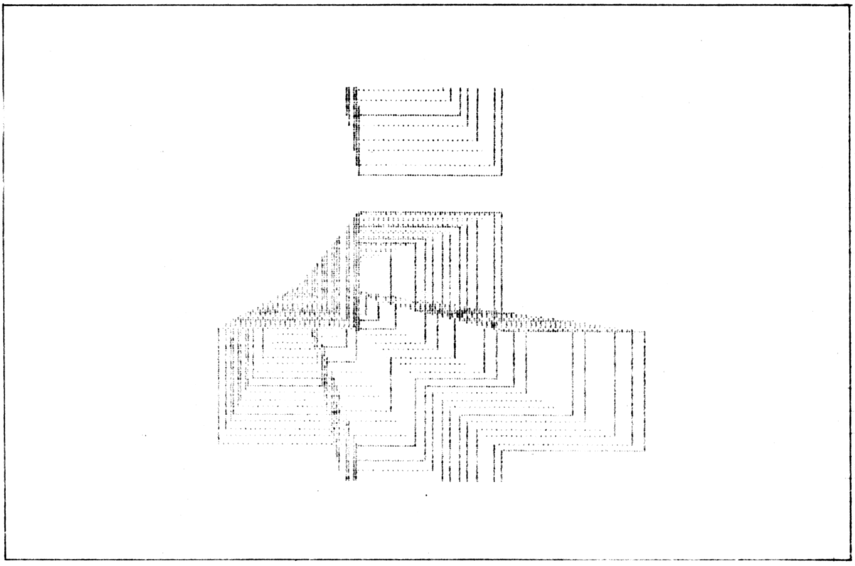


圖 1-11(d)

1-5 在APPLE的高解像畫面上作出文數字數字與圖形單元

APPLE Ⅱ這部微電腦的優點之一是它的彩色高解像度畫面（HIGH RESOLUTION GRAPHICS），在做函數圖形、曲線之描繪更見其特長，但會使用者常感不便的是在做圖形與文數字混合使用時，便沒有如ATARI，TRS-80 COLOR，PC-8001，VIC-20等的方便了，因此APPLE Ⅱ是一部相當有潛力的機器，但却需要使用者的靈巧安排。

在本程式裡頭，我們可以學到如何在高解像畫面上做出文字、數字，以及其它特殊的符號、圖形。讀者可將它和自己的程式配合使用以發揮更大的功能。因此，我們不要單獨以一個程式來看它，更要加以擴展、利用才是。本程式是以BASIC語言寫成的，如果要增加其執行速度的話，可嘗試以6502本身的機械語言配合APPLE監督程式（MONITOR）中的副程式來使用。

1. 畫面結構

在解析本程式以前，先要對APPLE的畫面結構加以瞭解，才能靈活使用。首先依照下列程式在APPLE上執行：

```
]LIST

5  HOME
10  FOR I = 0 TO 255
20  POKE 1024 + I,I
30  NEXT I

]RUN
```

我們可發現許多文、數字、符號在畫面上出現，而其出現的次序是在塡滿了畫面最上面的第 0 列之後却跳到第 8 列開始，然後再從第 16 列依次出現文字或圖形。由1-1節參照APPLE的記憶位址圖（Memory Map）可發現：在文字幕畫面時（TEXT MODE），其畫面位址是由第1024至2039的這個區間來組成的。請參考表1-1的位址及每行印出文字的次序，並注意從第 16 列之後每列的最後位址與下次掃描的開始位址並不連續，其間空了 8 個 RAM 的位址（第 16 列之後共有 8 列，每列 8 個位址，因此有 64 個位址沒使用到，APPLE使用了 1K 的位址空間來做字幕顯示，1K 有1024 個位址，1024－64＝960＝40 字× 24 行，正好是APPLE的字幕格式）。使用低解像度畫面時，其畫面印出格式也是一樣的。只不過每一文字位置是由兩個低解像方格來代替。

與字幕顯示方式不同的，高解像度的畫面上，RAM的每一位址是代表高解像度的七個點而不是一個字形，畫面上每一列共包含了40個RAM位址，因此共有 40 × 7 ＝ 280 個點。而每一位址裡包含的八個位元（Bit）中，最高位元並不被用來顯示，而是做爲顏色選擇用的，因此最低位元（第 0 位）排列在螢幕的最左邊，然後依次是第一、第二、……等位元，而第六位元之後就接下來是次一位址的第 0 位元了。關於顏色的使用規則，根據APPLE Ⅱ的參考手册（R1）指出：當一個點的相對應位元是“0”的時候表示該點不被顯示（即黑色），如果該點的位元值是“1”的時候，則依照下列原則：

⑴　在偶數行的點，如果其位址最高位元（即不被顯示的那一位元）是“0”則爲紫色。若最高位元爲“1”則爲藍色。

⑵　在奇數行的點，如果其位址最高位元是“0”則爲綠色。若最高位元爲“1”則爲橙色。

⑶　任何相鄰兩點若同時點亮顯示時，則顏色變爲白色。

由上面規則知道是不可能令同一位址上的點同時出現綠色及藍色，綠色及橙色，紫色及藍色或紫色及橙色的。

表 1-1 文字幕的畫面記憶位址

	A	B	C	D	
A	1024	0	0	1063	
B	1152	1	3	1191	
C	1280	2	6	1319	
D	1408	3	9	1447	
	1536	4	12	1575	
	1664	5	15	1703	
	1792	6	18	1831	
	1920	7	21	1959	
	1064	8	1	1103	
	1192	9	4	1231	
	1320	10	7	1359	
	1448	11	10	1487	
	1576	12	13	1615	
	1704	13	16	1743	
	1832	14	19	1871	
A：該列第一個位址	1960	15	22	1999	
B：該列在畫面上的次序	1104	16	2	1143	⋮
C：該列在 RAM 裡頭的	1232	17	5	1271	⋮
次序	1360	18	8	1399	⋮
D：該列最後位址	1488	19	11	1527	每列空下
	1616	20	14	1655	8 個位址
	1744	21	17	1783	⋮
	1872	22	20	1911	⋮
	2000	23	23	2039	⋮

仿照我們對字幕畫面與RAM記憶位址之間關係的解析方式，可在APPLE上執行下列程式：

```
]LIST

10   HGR
20   FOR I = 0 TO 8191
30   POKE 8192 + I,255
40   NEXT
```

這個程式的意思是：將255（十進位）的資料依次填入從8192這個位址起共 8K的位址，這 8K位址正好是高解像度上第一頁的整個畫面，而255在十六進位裡是 FF 亦即是將每一位址的所有位元設定爲“1”，依照上列第三項規則：是一連串的白點排列成的水平直線。現在觀察畫面水平直線依次出現的格式，請參考圖1-12所示。

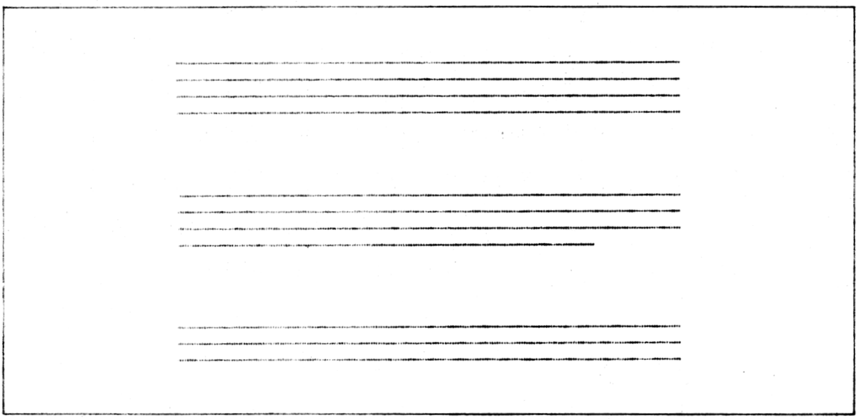


圖 1-12

我們可發現高解像度的圖像之位址對應方式和字幕顯示之位址對應方式類似，首先是把畫面依垂直方向分成三個區間依次掃描，這時每條線之間是空了七個點的間隔，等到畫面上三個區間填滿了之後再向下移了一個點的位置重新掃描，如此重覆了八遍之後就把畫面填滿了。全部畫面與記憶位址之安排可參照表1-2。

表 1-2 高解像畫面的記憶位址

0	8192- 8231	48	8960- 8999	96	8744- 8783	144	8528- 8567
1	9216- 9255	49	9984-10023	97	9768- 9807	145	9552- 9591
2	10240-10279	50	11008-11047	98	10792-10831	146	10576-10615
3	11264-11303	51	12032-12071	99	11816-11855	147	11600-11639
4	12288-12327	52	13056-13095	100	12840-12879	148	12624-12663
5	13312-13351	53	14080-14119	101	13864-13903	149	13648-13687
6	14336-14375	54	15104-15143	102	14888-14927	150	14672-14711
7	15360-15399	55	16128-16167	103	15912-15951	151	15696-15735
8	8320- 8359	56	9088- 9127	104	8872- 8911	152	8656- 8695
9	9344- 9383	57	10112-10151	105	9896- 9935	153	9680- 9719
10	10368-10407	58	11136-11175	106	10920-10959	154	10704-10743
11	11392-11431	59	12160-12199	107	11944-11983	155	11728-11767
12	12416-12455	60	13184-13223	108	12968-13007	156	12752-12791
13	13440-13479	61	14208-14247	109	13992-14031	157	13776-13815
14	14464-14503	62	15232-15271	110	15016-15055	158	14800-14839
15	15488-15527	63	16256-16295	111	16040-16079	159	15824-15863
16	8448- 8487	64	8232- 8271	112	9000- 9039	160	8784- 8823
17	9472- 9511	65	9256- 9295	113	10024-10063	161	9808- 9847
18	10496-10535	66	10280-10319	114	11048-11087	162	10832-10871
19	11520-11559	67	11304-11343	115	12072-12111	163	11856-11895
20	12544-12583	68	12328-12367	116	13096-13135	164	12880-12919
21	13568-13607	69	13352-13391	117	14120-14159	165	13904-13943
22	14592-14631	70	14376-14415	118	15144-15183	166	14928-14967
23	15616-15655	71	15400-15439	119	16168-16207	167	15952-15991
24	8576- 8615	72	8360- 8399	120	9128- 9167	168	8912- 8951
25	9600- 9639	73	9384- 9423	121	10152-10191	169	9936- 9975
26	10624-10663	74	10408-10447	122	11176-11215	170	10960-10999
27	11648-11687	75	11432-11471	123	12200-12239	171	11984-12023
28	12672-12711	76	12456-12495	124	13224-13263	172	13008-13047
29	13696-13735	77	13480-13519	125	14248-14287	173	14032-14071
30	14720-14759	78	14504-14543	126	15272-15311	174	15056-15095
31	15744-15783	79	15528-15567	127	16296-16335	175	16080-16119
32	8704- 8743	80	8488- 8527	128	8272- 8311	176	9040- 9079
33	9728- 9767	81	9512- 9551	129	9296- 9335	177	10064-10103
34	10752-10791	82	10536-10575	130	10320-10359	178	11088-11127
35	11776-11815	83	11560-11599	131	11344-11383	179	12112-12151
36	12800-12839	84	12584-12623	132	12368-12407	180	13136-13175
37	13824-13863	85	13608-13647	133	13392-13431	181	14160-14199
38	14848-14887	86	14632-14671	134	14416-14455	182	15184-15223
39	15872-15911	87	15656-15695	135	15440-15479	183	16208-16247
40	8832- 8871	88	8616- 8655	136	8400- 8439	184	9168- 9207
41	9856- 9895	89	9640- 9679	137	9424- 9463	185	10192-10231
42	10880-10919	90	10664-10703	138	10448-10487	186	11216-11255
43	11904-11943	91	11688-11727	139	11472-11511	187	12240-12279
44	12928-12967	92	12712-12751	140	12496-12535	188	13264-13303
45	13952-13991	93	13736-13775	141	13520-13559	189	14288-14327
46	14976-15015	94	14760-14799	142	14544-14583	190	15312-15351
47	16000-16039	95	15784-15823	143	15568-15607	191	16336-16375

2. 高解像畫面位置的計算

在上節中我們已計算出 RAM 記憶位址的每一位元在畫面上的對應位置，但是每次都要查表1-2也未免太麻煩了，因此，我們接下來要發展出一套方便的計算公式，使用這公式，可以將畫面上的位置座標（X，Y）換算成相對應的記憶位址，只要使用者輸入所欲設定的畫面 X－Y 位置，該程式即自動換成記憶空間的位址而加以處理。

首先參照APPLE REFERENCE MANUAL上第 21 頁的表格（見表1-3），該表的格式與字幕顯示時的格式（見表1-4）是一樣的，共有24×40 個方格，只不過在表1-3的每一方格裡，不是特定的字形而是高解像的點所組成的，每一方格包含了水平 7 個點垂直 8 個點的位置，因此垂直的24 個方格×8 點與水平的40 個方格×7 點構成了APPLE高解像度的280×192 的畫面。

表1-3　高解像度畫面結構

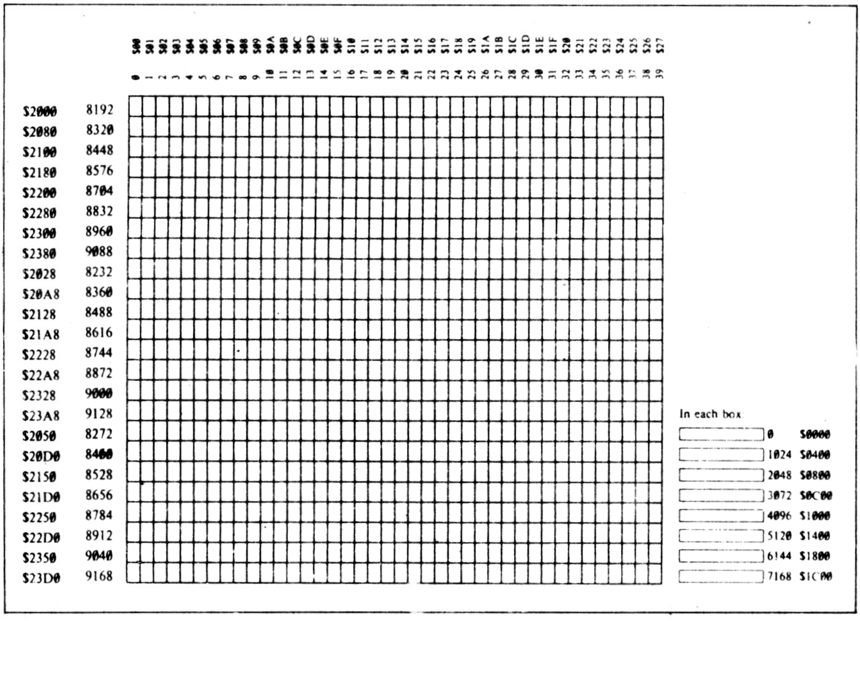
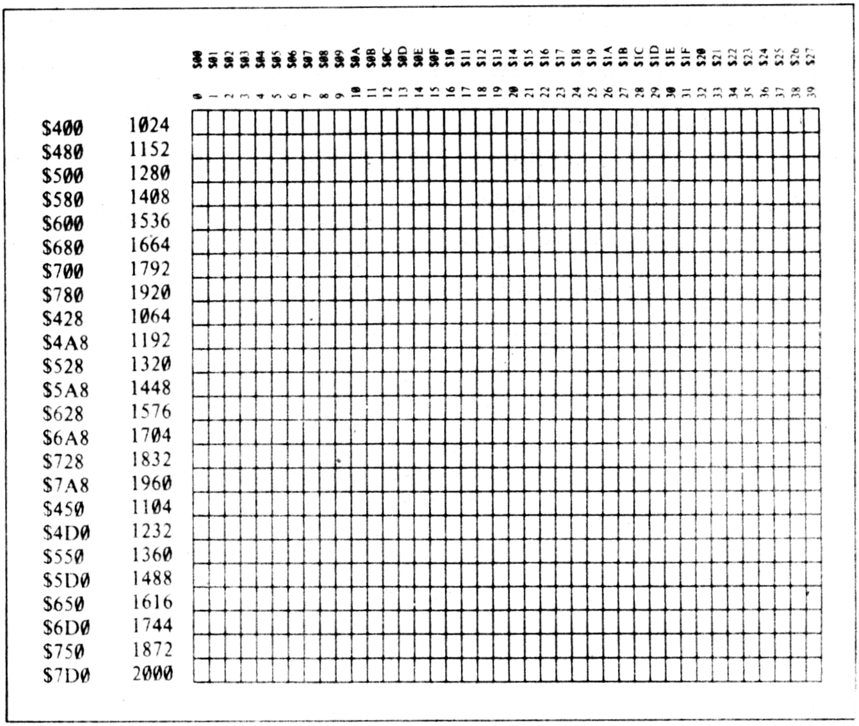


表1-4 文字幕的畫面結構



如果將表1-2與表1-3加以對照，譬如 $2484（ $ 表示十六進位）這個位址可依下示來算出其畫面上相對應的位置：

$2484=$2080+$0400+$04

這個式子代表由表1-3左邊的縱軸座標上找到 $2080 所代表的第二橫列方格，再由方格內找出 $0400 代表是方格裡的第二條掃描線，而$04即是由左邊算起第五個方格裡頭的7個點，因此 $2484 這個位址正是由（29,9）至（35,9）間7個點的（X,Y）座標了（注意：在第一橫列方格中共有8條掃描線，因此第二橫列方格的第二條掃描線應該是由畫面頂端算起的第十個位置，但座標是由0算起，所以Y的座標值是9，這裡所指的X－Y是192×280的高解像畫面）。

現在已經知道如何將記憶位址換成畫面上的X－Y座標了，至於如何反過來將X－Y座標值換成記憶位址呢？因爲在程式執行時常要做這種換算，所以請參照表1-2來觀察，由於畫面之格式是將畫面分成三等份，然後每等份分成八列，每列再由八條掃描線組成，在表1-2中我們可看出第0條掃描線與第64條掃描線及第64條掃描線與第128條掃描線之間均相隔了每條掃描線所使用的40個位址，所以第128條掃描線的開頭位址就是第0條掃描線的開始位址加上40個位址×2 即 8192＋40×2＝8272，再觀察相鄰每一橫列方格的對應掃描線之位址是相差128個位址，譬如第18列的第3條線的開始位址爲11600而第19列的第3條掃描線位址是11728，11728－11600＝128（請注意所有的列與線均是由第0號算起的）。瞭解了這個方式之後就可寫下記憶位址與X－Y畫面座標的關係了（注意：此處所指的X－Y是指24×40的字幕畫面）。

```
L=8192+Y MOD 8*128+Y/8*40+X

                X/Y==>INT(X/Y)
IN APPLESOFT :
                X MOD Y==>X-INT(X/Y)*Y
```

上式中，L代表記憶位址由8192位址算起，（X，Y）代表表1-3及表1-4中方格在畫面上的座標位置，這個公式是以Integer BASIC寫成的，如果是使用Applesoft BASIC時可依所列方式改寫。爲了更明白這個公式的意思，可擧個例子來檢視一下，以（X，Y）＝（3，19）爲例，19MOD8＝3，INT（19/8）＝2，亦即L＝8192＋3×128＋2×40＋3＝8659，由表1-2查出第19行第0個方格的位址是8656，所以第3個方格的位址正是8659。爲了要使在高解像畫面上所產生出來的文字、數字之大小、格式要與文字幕顯示（TEXT MODE）時一致，因此採用X由0～39，Y由0～23的格式。

3. 字形與圖案的產生

在瞭解了表1-2及表1-3的方格在高解像畫面上的位置關係之後，現

在進一步來瞭解每一方格內的情形。每一方格在水平方向是由每一位址除了最高位元的其它 7 個位元所組成，因此共有 7 個點，而點的顏色是由前述的規則來決定的。方格垂直方向是由 8 條掃描線組成，每條掃描線的長度是一個位址的 7 個位元，因此只要由程式來設定方格內的每一位址的內容（即位元的組合），就可產生各種 7 × 8 的點陣字形或圖案了，如果爲了要和 APPLE 上的字幕顯示（TEXT MODE）一致的話，則可令每一方格的最底線所有位元爲“0”而其它 7 條線的第 0 位元與第六位元亦爲“0”，如此則構成一 5 × 7 的點陣字形或圖案了，請參照圖 1-13 。

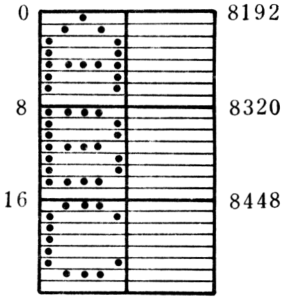


圖 1-13

因此，將字形產生出來，再給予畫面的 X，Y 座標值就可以將該字形置於所欲放置的位置了。由圖 1-13 的字形掃描，可看出每個 5 × 7 的點陣字形共需 8 個 RAM 的記憶位址，8192 這個位址正好是畫面最頂端第“0”號掃描線的最左端 7 個點（位元），在圖 1-13 的例子裡正好將“A”這個字的最頂端一點點亮起來，也就是第 3 個位元是“1”，而其他位元的值是“0”，也可以說 8192 這個位址的內含值（資料）是 8 。但是 8193 這個位址並不代表“A”這個字的第二條掃描線，而是沿著第 0 號掃描線向右接著排列過去，要過了 128 個位址之後才又接著掃描“B”這個字的頂端掃描線（也就是畫面上的第 8 號掃描線），這時的位址是 8320 ，在圖 1-13 中它的內含值是 30（十進位）。圖 1-13 中的 ABC 三個字形左邊的數是代表掃描線的號數，而右邊的數字是代表其位址。

那麼，如何來造出所需要的字形呢？由圖1-13來觀察，“A”這個字形的8條掃描線其位址分別可由表1-2來查出即：8192，9216，10240，11264，12288，13312，14336及15360等8個RAM的記憶位址（注意：每個位址間正好差1024個位址）。而其每條掃描線的形狀（即點的亮暗）亦即這8個RAM位址的內含值，因此分別將8，20，34，34，62，34，34，0依次存入上述的八個位址，就可在畫面的左上角看到一個“A”字了。這些內含值是以二進位法依圖1-13來排列的。如果我們執行POKE 15360，62這道指令，就會在“A”字的第7劃掃描線上點亮連續的五個點（62＝2＋4＋8＋16＋32）（所以看到的是白色的線）。

0	1	2	3	4	5	6	7
1	2	4	8	16	32	64	128
LSB							MSB

圖1-14

以上是純由字形上的考慮，如果要做顏色的設定，則要將最高位元（MSB）亦做詳細規劃才行，可由程式設計者自行訂定之。所以如何在畫面上造出各種字形就很容易明白了，首先將每個字形（方格）的8條掃描線以A、B、C、D、E、F、G、H來代表，而其值即掃描線的形狀（即各點的暗亮），然後將該字形的位置（X，Y）座標定出，由上述的公式換成記憶位址L，這就是該字形最頂端掃描線的位址，而次一掃描線的位址是L＋1024，L＋2048，……，依此類推，所以將A、B、C、D、……等之值POKE進入L、L＋1024、L＋2048、……等就可以將所要的字形顯現在高解像畫面上的第Y列第X個位置了。

4. 程式說明

程式A是以APPLESOFT寫成的，由行號110到173是文字及符號的資料，其行號分別代表各文字、符號的ASCII碼，在APPLE上的

ASCII 碼是標準ASCII 碼加上128 而成的，譬如“A”的ASCII 碼是65＋128＝193 ，因第143 行的資料就是“A”的資料即A＝8 、B＝20 、C＝34 、D＝C 、E＝62 、F＝C 、G＝C ，所以行號與ASCII 碼之間的差額是 50 。

程式由第10 行開始，在顯示高解像全畫面並將之消除之後，GOTO 第1000行，然後經過 173 及 100 兩道副程式，前者產生游標，後者將游標置於畫面的左上角也就是8192 的位址（因為此時X與Y均未設定，根據APPLE的規則其值為零）。

由173 及100 行的副程式回來後，接着執行第1010行，此行首先以K＝PEEK（－16384 ）來檢知鍵盤所按的鍵的 ASCII 值，如果所按的鍵不是“←”或“BS”（其ASCII 值是136 ），則繼續往下執行第1020行，如果K＝136 ，則到110行將字形資料設定為空白並印在先前的 X，Y 座標上，因此前一個字就被消去了，如果恰巧是在一行的最左端，則其前一個字是上一行的最後一個字而Y＝Y－1 ，X＝X＋40（即第1015行所執行的）。

第1020行檢視是否所按的鍵是“CR”（或RETURN），“CR”的 ASCII 值是141 ，若是則將游標移往下一行的開頭，若Y已是第23行（最後一行）則自動跳往第一行。

第1030行檢視Z是否小於160 ，160 以下屬於控制碼範圍，不是代表字形，在本程式中不使用，若不是則到K所屬的行號去取得K字形的資料，然後到第100行將之印出（記得K就是所按鍵的 ASCII 碼）。

第1050行將X的值加一，亦即向右移一格，再到第1000行等待按鍵，若X已是最右的一格，則將游標移至下一行的最左邊，若已是在最底下的一行的最後一個字，則游標移到左上角的位置。

程 式：

```
]LIST

10  HGR : POKE  - 16302,0: GOTO 100
     0
100 W = Y -  INT (Y / 8) * 8:L = 81
     92 + W * 128 +  INT (Y / 8) *
     40 + X
105  POKE L,A: POKE L + 1024,B: POKE
     L + 2048,C: POKE L + 3072,D: POKE
     L + 4096,E: POKE L + 5120,F: POKE
     L + 6144,G: POKE L + 7160,0
106  RETURN
110 A = 0:B = A:C = A:D = A:E = A:F
      = A:G = A: RETURN
111 A = 8:B = A:C = A:D = A:E = A:F
      = 0:G = A: RETURN
112 A = 20:B = A:C = A:D = 0:E = D:
     F = D:G = A: RETURN
113 A = 20:B = A:C = 62:D = A:E = C
     :F = A:G = A: RETURN
114 A = 8:B = 60:C = 10:D = 28:E =
     40:F = 30:G = A: RETURN
115 A = 6:B = 38:C = 16:D = 8:E = 4
     :F = 50:G = 48: RETURN
116 A = 4:B = 10:C = 8:D = A:E = 42
     :F = 18:G = 44: RETURN
117 A = 8:B = A:C = A:D = 0:E = D:F
      = D:G = D: RETURN
118 A = 8:B = 4:C = 2:D = C:E = C:F
      = B:G = A: RETURN
```

```
119 A = 8:B = 16:C = 32:D = C:E = C
    :F = B:G = A: RETURN
120 A = 8:B = 42:C = 28:D = A:E = C
    :F = B:G = A: RETURN
121 A = 0:B = 8:C = B:D = 62:E = B:
    F = B:G = A: RETURN
122 A = 0:B = A:C = A:D = A:E = 8:F
     = E:G = 4: RETURN
123 A = 0:B = A:C = A:D = 62:E = A:
    F = A:G = A: RETURN
124 A = 0:B = A:C = A:D = A:E = A:F
     = A:G = 8: RETURN
125 A = 0:B = 32:C = 16:D = 8:E = 4
    :F = 2:G = A: RETURN
126 A = 28:B = 34:C = 50:D = 42:E =
    38:F = B:G = A: RETURN
127 A = 8:B = 12:C = A:D = A:E = A:
    F = A:G = 28: RETURN
128 A = 28:B = 34:C = 32:D = 24:E =
    4:F = 2:G = 62: RETURN
129 A = 62:B = 32:C = 16:D = 24:E =
    B:F = 34:G = 28: RETURN
130 A = 16:B = 24:C = 20:D = 18:E =
    62:F = A:G = A: RETURN
131 A = 62:B = 2:C = 30:D = 32:E =
    D:F = 34:G = 28: RETURN
132 A = 56:B = 4:C = 2:D = 30:E = 3
    4:F = E:G = 28: RETURN
133 A = 62:B = 32:C = 16:D = 8:E =
    4:F = E:G = E: RETURN
134 A = 28:B = 34:C = B:D = A:E = B
    :F = B:G = A: RETURN
```

```
135 A = 28:B = 34:C = B:D = 60:E =
     32:F = 16:G = 14: RETURN
136 A = 0:B = A:C = 8:D = A:E = C:F
     = A:G = A: RETURN
137 A = 0:B = A:C = 8:D = A:E = C:F
     = C:G = 4: RETURN
138 A = 16:B = 8:C = 4:D = 2:E = C:
     F = B:G = A: RETURN
139 A = 0:B = A:C = 62:D = A:E = C:
     F = A:G = A: RETURN
140 A = 4:B = 8:C = 16:D = 32:E = C
     :F = B:G = A: RETURN
141 A = 28:B = 34:C = 16:D = 8:E =
     D:F = 0:G = D: RETURN
142 A = 28:B = 34:C = 42:D = 58:E =
     26:F = 2:G = 60: RETURN
143 A = 8:B = 20:C = 34:D = C:E = 6
     2:F = C:G = C: RETURN
144 A = 30:B = 34:C = B:D = A:E = B
     :F = B:G = A: RETURN
145 A = 28:B = 34:C = 2:D = C:E = C
     :F = B:G = A: RETURN
146 A = 30:B = 34:C = B:D = B:E = B
     :F = B:G = A: RETURN
147 A = 62:B = 2:C = B:D = 30:E = B
     :F = B:G = A: RETURN
148 A = 62:B = 2:C = B:D = 30:E = B
     :F = B:G = B: RETURN
149 A = 60:B = 2:C = B:D = B:E = 50
     :F = 34:G = A: RETURN
150 A = 34:B = A:C = A:D = 62:E = A
     :F = A:G = A: RETURN
```

```
151 A = 28:B = 8:C = B:D = B:E = B:
    F = B:G = A: RETURN
152 A = 32:B = A:C = A:D = A:E = A:
    F = 34:G = 28: RETURN
153 A = 34:B = 18:C = 10:D = 6:E =
    C:F = B:G = A: RETURN
154 A = 2:B = A:C = A:D = A:E = A:F
     = A:G = 62: RETURN
155 A = 34:B = 54:C = 42:D = C:E =
    A:F = A:G = A: RETURN
156 A = 34:B = A:C = 38:D = 42:E =
    50:F = A:G = A: RETURN
157 A = 28:B = 34:C = B:D = B:E = B
    :F = B:G = A: RETURN
158 A = 30:B = 34:C = B:D = A:E = 2
    :F = E:G = E: RETURN
159 A = 28:B = 34:C = B:D = B:E = 4
    2:F = 18:G = 44: RETURN
160 A = 30:B = 34:C = B:D = 30:E =
    10:F = 18:G = B: RETURN
161 A = 28:B = 34:C = 2:D = A:E = 3
    2:F = B:G = A: RETURN
162 A = 62:B = 8:C = B:D = B:E = B:
    F = B:G = B: RETURN
163 A = 34:B = A:C = A:D = A:E = A:
    F = A:G = 28: RETURN
164 A = 34:B = A:C = A:D = A:E = A:
    F = 20:G = 8: RETURN
165 A = 34:B = A:C = A:D = 42:E = D
    :F = 54:G = A: RETURN
166 A = 34:B = A:C = 20:D = 8:E = C
    :F = A:G = A: RETURN
```

```
167 A = 34:B = A:C = 20:D = 8:E = D
     :F = D:G = D: RETURN
168 A = 62:B = 32:C = 16:D = 8:E =
     4:F = 2:G = A: RETURN
169  RETURN
170  RETURN
171 A = 62:B = 48:C = B:D = B:E = B
     :F = B:G = A: RETURN
172 A = 0:B = A:C = 8:D = 20:E = 34
     :F = A:G = A: RETURN
173 A = 0:B = A:C = A:D = A:E = A:F
      = A:G = 28: RETURN
1000  GOSUB 173: GOSUB 100
1010 K =  PEEK ( - 16384): IF K < 1
     28 THEN 1010
1011  POKE  - 16368,0: IF K <  > 13
     6 THEN 1020
1012  GOSUB 110: GOSUB 100:X = X -
     1
1015  IF X < 0 THEN Y = Y - 1
1016  IF X < 0 THEN X = X + 40
1017  IF Y < 0 THEN Y = Y + 24
1018  GOTO 1000
1020  IF K <  > 141 THEN 1030
1021  GOSUB 110: GOSUB 100:X = 0:Y =
     Y + 1: IF Y > 23 THEN Y = 0
1022  GOTO 1000
1030  IF K < 160 THEN 1010
1031  IF K > 200 THEN 1033
1032 H = K - 159: ON H GOSUB 110,11
     1,112,113,114,115,116,117,118,
     119,120,121,122,123,124,125,12
```

```
       6,127,128,129,130,131,132,133,
       134,135,136,137,138,139,140,14
       1,142,143,144,145,146,147,148,
       149,150: GOTO 1034
1033 H = K - 200: ON H GOSUB 151,15
       2,153,154,155,156,157,158,159,
       160,161,162,163,164,165,166,16
       7,168,169,170,171,172,173
1034  GOSUB 100
1050 X = X + 1: IF X < 40 THEN 1000

1051 X = X - 40:Y = Y + 1: IF Y < 2
       4 THEN 1000
1052 Y = Y - 24: GOTO 1000
```

執行例:

在某些需要文字與圖形混合的場合，將本程式加以應用就可達成目的。另外，將本程式擴充，可以增加些常用的圖形，而以 SHIFT 鍵來控制（稱爲圖案字元），如此可使圖形製作增添不少方便。如圖 1-15 所示，即爲一將文字與圖形混合的例子。

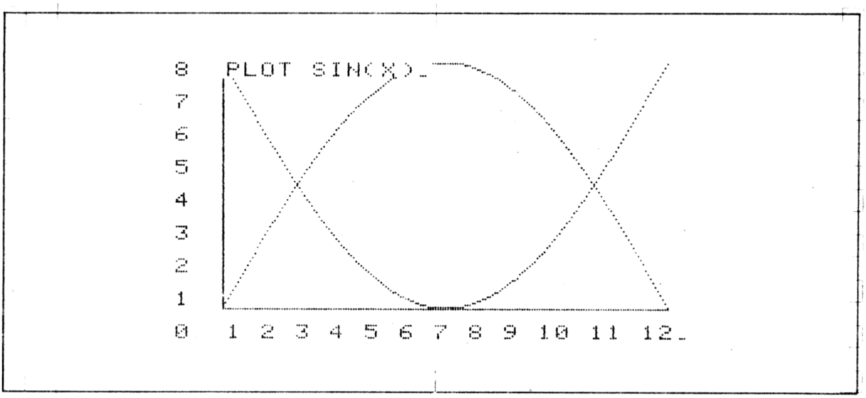


圖 1-15

第二部份

APPLE Ⅱ程式集

在這一部份，我們收集一些有趣而具代表性的程式，其中一些遊戲（GAME）程式是爲了令讀者在研究當中也能產生熱烈的興趣。在收集這些程式的時候，特別注意到如何將 APPLE Ⅱ 所具有的特性加以發揮，也就是說把它獨特的性能加以表現，以便程式設計者能注意到這種優點而善加運用，而發揮個人電腦（ RERSONAL COMPUTER ）的特性所在。

雖然是一些簡單的程式，可是却方便於透徹的明瞭其運作情形。同時若要加以引用成爲其他程式的一部份時，就更加方便。讀者可以嘗試這種程式的整合工作（ APPEND ）。如果這些小程式可以給你任何啓示而加以發展的話，則未嘗不能構成更大而優秀的作品的。

程式一 不會停止的圖案

當我們經過經售微電腦商店的櫥窗時，常會看到許多美麗的電腦圖案，這些美麗而多樣變化的圖案是怎麼做成的呢？除了在本書第一部份1-4節中有關圖形表的例子外，在這裡另外舉出一個例子，以簡短的 20 個敍述就可以做出有趣的圖案來（請與APPLE使用手册BASIC PROGRAMING REFERENCE MANUAL 第 26 頁的程式比較）。

程 式

```
]LIST

120  HGR2
125 M% =  INT ( RND (2.78) * 5) + 1

130  FOR M = 1 TO M%
131 P% =  RND (4) * 6 + 1
132  IF P% = 4 THEN 131
133 O% =  RND (4) * (7 - P%)
134  IF P% = 3 THEN  IF O% = 1 THEN
     133
135  IF (P% + O%) > 7 THEN 131
180 I% = ( RND (1) * 8) + 2
220  FOR X = 0 TO 278 STEP I%
```

```
240  FOR S = 0 TO 1
260  HCOLOR= P% * S + Q%
300  HPLOT X + S,0 TO 140,95 TO 279
      - X - S,191
320  NEXT S,X
321  FOR X = 190 TO 0 STEP  - I% *
     1.01
322  FOR S = 0 TO 1
323  HCOLOR=  - P% * S + (P% + Q%)
324  HPLOT 0,X + S TO 140,95 TO 279
     ,191 - X - S
325  NEXT S,X
330  NEXT M: GOTO 120
```

程式說明

由第120行設定畫面爲HIRES（高解像度的簡稱）的第二頁區。之後，由第125行、第131行、第133行及第135行，利用亂數產生函數RND 來設定M%、P%、Q%及I%等任意整數，其中M%是用做第130行及第330行的FOR-NEXT 廻路（LOOP）用，當M%的廻路走完之後，畫面由HGR2 重新消除，程式從頭開始。I%是用以決定圖中畫線端點的間隔，即第220行及第321行中的STEP I%（參照圖2-1）。

第131行及第133行的P%及Q%用以設定第260行及第323行的HCOLOR 的顏色，此二行另外由S＝0及1來設定HCOLOR ＝ P%＋Q%或Q%，其主要的目的是爲了使圖樣顏色多變化。第132行及第134行是避免使P%及P%＋Q%＝4，因爲第4色爲黑色，雖然第0色也是黑色，但已由程式設定Q%、P%＋Q%、P%等不會爲零。

畫圖的過程是由第220行至第320行來畫出圖2-2中的上下兩個三角形，而第321行至第325行來畫出左右兩個三角形，所有這些線均經過中央點座標（140，95），可參照圖2-1。

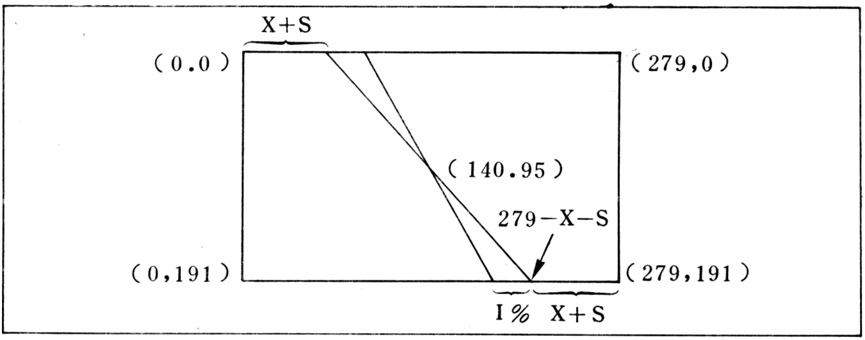


圖 2-1

至於在圖2-2裡，也就是螢幕上看到的整個圖案，除了是由於顏色不同所造成的以外，同時還利用了APPLE的高解像畫面的特性，也就是說除了白色和黑色以外，其他顏色有的只能畫在水平線方向是奇數的點，有的只能畫在是偶數的點，如此使得圖案更形多樣變化而美麗了。至於為何只能畫在奇數或偶數點上，請參考R1、R6的參考資料。

執行例

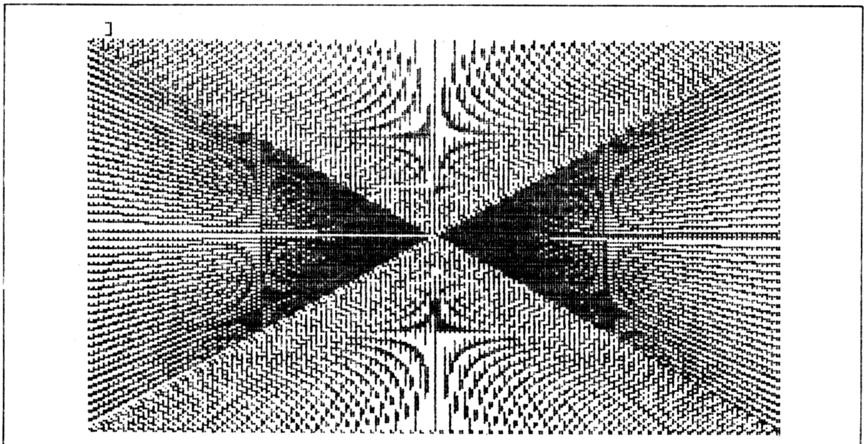


圖 2-2

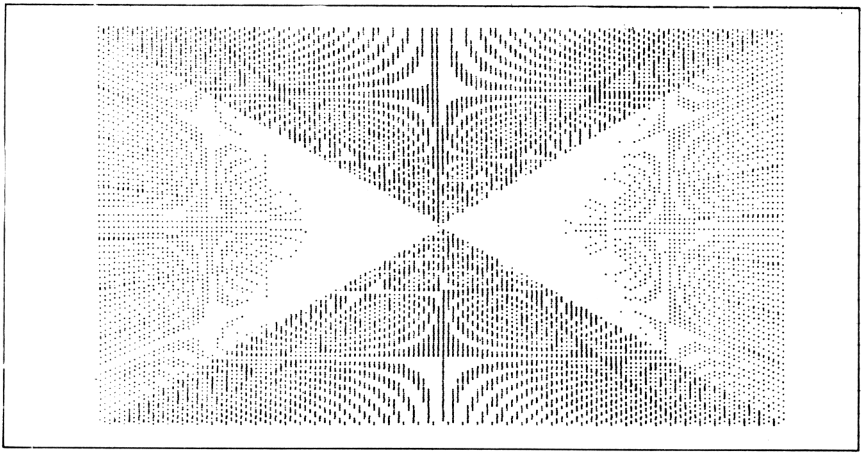


圖 2-3

程式二 美麗的彩色文字

這是利用APPLE 16色的低解像畫面來產生各種彩色的字形符號，除了因此而熟悉程式編寫的一些技巧外，也可藉此機會來瞭解APPLE的畫面結構，另外對於字形組合也能有更清晰的瞭解。

對於同樣的目的，我們分別以兩種不同的方式來寫了兩個程式，並分別列出加以比較，我們也可以經由這種比較來瞭解電腦本身程式運轉的情況，以便對於以後設計上的參考與判斷，所以這則程式可說是其學習研究性質較諸實用性爲高，但是巧妙的將本程式附加在其他主要程式上，却更可發揮意想不到的功能，希望能善加利用。

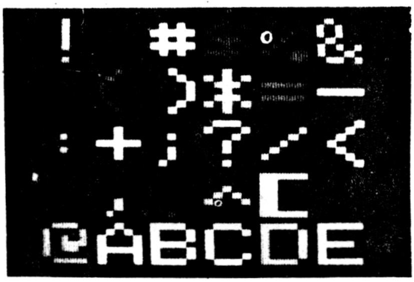


圖 2-4 由本程式作出五顏六色的美麗字形

畫面結構

衆所週知的，APPLE的畫面分成高、低兩種解像度，其中低解像度是和文字幕共用一個記憶區的，這個區域是從$400至$7FF的範圍（亦

即十進位的1024至2047的區間），依程式指令的要求，透過軟體開關（SOFT SWITCH）使用者可選擇各種畫面及文字顯示的形式。

文字幕的格式是24 列× 40 字，而低解像畫面則是 48 列× 40 格或 40 列× 40 格加上底部 4 行文字幕，因此可見低解像度的方格單元個數恰為文字幕的兩倍，而且每一方格寬度與文字幕時大小相同，但是其高度却是字形的一半而已，讀者可參考表 1-4 及表2-1，在此我們將文字幕及低解像畫面的方格座標圖印出來，以方便程式設計時可在表格上製作出圖形來。

表 2-1 低解像度的畫面記憶位址結構

		$00	$01	$02	$03	$04	$05	$06	$07	$08	$09	$0A	$0B	$0C	$0D	$0E	$0F	$10	$11	$12	$13	$14	$15	$16	$17	$18	$19	$1A	$1B	$1C	$1D	$1E	$1F	$20	$21	$22	$23	$24	$25	$26	$27
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
$400	1024																																								
$480	1152																																								
$500	1280																																								
$580	1408																																								
$600	1536																																								
$680	1664																																								
$700	1792																																								
$780	1920																																								
$428	1064																																								
$4A8	1192																																								
$528	1320																																								
$5A8	1448																																								
$628	1576																																								
$6A8	1704																																								
$728	1832																																								
$7A8	1960																																								
$450	1104																																								
$4D0	1232																																								
$550	1360																																								
$5D0	1488																																								
$650	1616																																								
$6D0	1744																																								
$750	1872																																								
$7D0	2000																																								

字形的編排

參照圖2-5所示，英文字或數字符號可以用 5 × 7 的矩陣排列出來，圖中要畫出來的點以 1 表示，不畫出來的以 0 表示，所以“ A ”這個字是由“ 00100，01010，10001，10001，11111，10001 ，10001 ”構成的，如果所要畫出的是螢光幕上的每一掃描點（譬如在高解像畫面

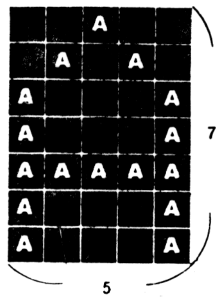


圖 2-5 字形"A"的 5 × 7 結構

裡），則這些 1 與 0 就代表記憶單元每一位元（bit）的 ON 或OFF，但是在低解像畫面情況就不同了，爲了要將這些點表示出來，我們可以將 0 用黑色（COLOR＝0）來表示，而 1 則用有顏色的點畫出來。

因此我們利用下列的方式來處理這個問題。首先，在 BASIC 裡像"10001"這個數是以十進位來看待的，因此要將此數的每一位數分別來判斷其爲 0 或 1，以便決定是否要畫出一個點來。所以利用整數變的 P％（J）把此數字由DATA 欄讀進來，然後以A＝P％（J）/10將P％（J）的最低位變成小數點的第一位，再令P％（J）＝A，此時的P％（J）已經是比原來的值小 10 倍並且再整數化了，因爲DATA的每一位數不是 0 便是 1，所以令M＝A－P％（J），則M不是 0 便是0.1（讀者可自行筆算試試），最後用"IF M THEN"的指令來決定如果M是0.1則畫出一點（即代表該位數爲 1），如果M是 0 則畫出黑色的點（因若最小位爲 0，則A＝P％（J）/10恰好整除，因此M＝0）。由此我們就可畫出一個美麗的字形來了。

但是，由於上述的位數判別（即 0 或 1）是由右邊的最低位數開始，而畫圖時是由左而右的，所以畫出來的字恰好是像照鏡子般的左右相對調，爲了要解決這個問題，必須把字形資料做鏡射處理，譬如圖2-6的"B"字第一列原來是"11110"的資料必須改成"01111"才行。

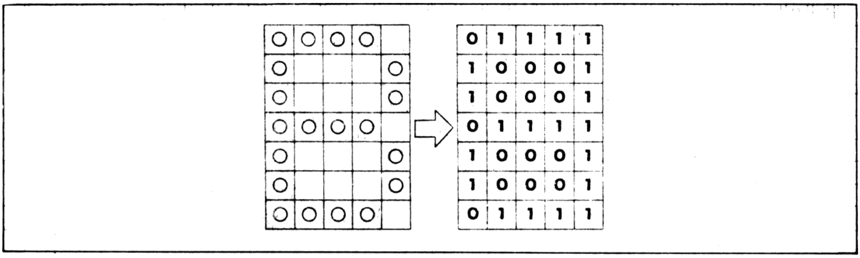


圖 2-6　將字形換成DATA

程　式

```
] LIST

100  DATA  0,0,0,0,0,0,0:REM
110  DATA  100,100,100,100, 100 ,0 ,100:
     REM!
120  DATA  1010,1010,0,0,0,0,0: REM "
130  DATA  0,1010,11111,1010,11111,1010,
     0:REM #
140  DATA  100,11110,101 ,1110,10100,1111,
     100:REM $
150  DATA  11,10011,1000, 100 ,10 , 11001 ,
     11000:REM %
160  DATA  10,101,101,10,10101,1001,10110
     :REM &
170  DATA  100,100,100,0,0,0,0:REM ,
180  DATA  100,10,1 ,1 ,1,10,100:REM (
190  DATA  100,1000,10000 , 10000 , 10000 ,
     1000,100:REM )
200  DATA  100,10101,1110,100,1110,10101
     ,100:REM *
```

```
210 DATA 0,100,100,11111,100,100,0:
    REM +
220 DATA 0,0,0,0,100,100,10:REM ,
230 DATA 0,0,0,11111,0,0,0:REM
240 DATA 0,0,0,0,0,0,100:REM ·
250 DATA 0,10000,1000,100,10,1,0:
    REM /
260 DATA 1110,10001,11001,10101,10011,
    10001,1110:REM 0
270 DATA 100,110,100,100,100,100,1110
    :REM 1
280 DATA 1110,10001,10000,1100.10,1,
    11111:REM 2
290 DATA 11111,10000,1000,1100,10000,
    10001,1110:REM 3
300 DATA 1000,1100,1010,1001,11111,
    1000,1000:REM 4
310 DATA 11111,1,1111,10000,10000,10001
    ,1110:REM 5
320 DATA 11100,10,1,1111,10001,10001,
    1110:REM 6
330 DATA 11111,10000,1000,100,100,100,
    100:REM 7
340 DATA 1110,10001,10001,1110,10001,
    10001,1110:REM 8
350 DATA 1110,10001,10001,11110,10000,
    1000,111:REM 9
360 DATA 0,0,100,0,100,0,0:REM :
```

```
370 DATA 0,0,100,0,100,100,10:REM ;
380 DATA 10000,1000,100,10,100,1000,
    10000:REM <
390 DATA 0,0,11111,0,11111,0,0:REM =
400 DATA 10,100,1000,10000,1000,100,
    10:REM >
410 DATA 1110,10001,1000,100,100,0,100
    :REM ?
420 DATA 1110,10001,10101,11101,1101,
    1,11110:REM @
430 DATA 100,1010,10001,1000 ,11111,
    10001,10001:REM A
440 DATA 1111,10001,10001,1111,10001,
    10001,1111:REM B
450 DATA 1110,10001,1,1,1,10001,1110:
    REM C
460 DATA 1111,10001,10001,10001,10001
    ,10001,1111:REM D
470 DATA 11111,1,1,1111,1,1,11111:REM E
480 DATA 11111,1,1,1111,1,1,1:REM F
490 DATA 11110,1,1,1,11001,10001,11110
    :REM G
500 DATA 10001,10001,10001,11111,10001
    ,10001,10001:REM H
510 DATA 1110,100,100,100,100,100,1110
    :REM I
520 DATA 10000,10000,10000,10000,10000
    ,10010,1100:REM J
```

```
530 DATA 10001,1001,101,11,101,1001,
    10001:REM K
540 DATA 1,1,1,1,1,1,11111:REM L
550 DATA 10001,11011,10101,10101,10001,
    10001,10001:REM M
560 DATA 10001,10001,10011,10101,11001,
    10001,10001:REM N
570 DATA 1110,10001,10001,10001,10001,
    10001,1110:REM O
580 DATA 1111,10001,10001,1111,1,1,1
    :REM P
590 DATA 1110,10001,10001,10001,10101,
    1001,10110:REM Q
600 DATA 1111,10001,10001,1111,101,1001
    ,10001:REM R
610 DATA 1110,10001,1,1110,10000,10001
    ,1110:REM S
620 DATA 11111,100,100,100,100,100,100
    :REM T
630 DATA 10001,10001,10001,10001,10001
    ,10001,1110:REM U
640 DATA 10001,10001,10001,10001,10001
    ,1010,100:REM V
650 DATA 10001,10001,10001,10101,10101
    ,11011,10001:REM W
660 DATA 10001,10001,1010,100,1010,10001
    ,10001:REM X
670 DATA 10001,10001,1010,100,100,100,
```

```
    100：REM Y
680 DATA 11111，10000，1000，100，10，1，
    11111：REM Z
690 DATA 11111，11000，11000，11000，11000
    ，11000，11111：REM ]
700 DATA 0，10000，1000，100，10，1，0：REM
    BACK SLASH
710 DATA 11111，11，11，11，11，11，11111：REM [
720 DATA 0，0，100，1010，10001，0，0：REM ∧
730 DATA 0，0，0，0，0，0，11111：REM
    UNDERLINE
740 HOME：L＝0：C＝L：GR
750 GET A$：IF A$＝"  "THEN TEXT：HOME：
    END
760 N＝ASC(A$)－31：RESTORE：FOR I＝1 TO
    N：READ P%(0)，P%(1)，P%(2)，P%(3)，
    P%(4)，P%(5)，P%(6)：NEXT：IF C＜35
    THEN 780
770 C＝0：L＝L＋8：IF L＞39 THEN L＝0：C＝L
780 R＝INT(15＊RND(1)＋1)
790 FOR J＝0 TO 6：FOR I＝0 TO 4：A＝P%
    (J)/10：P%(J)＝A：M＝A－P%(J)：IF M THEN
    COLOR＝R：PLOT C＋I，L＋J：GOTO 810
800 COLOR＝0：PLOT C＋I，L＋J
810 NEXT：NEXT：C＝C＋6：GOTO 750
```

程式說明

在上列的程式中，從第100行至第730行均為字形資料，而從第740

行之後才是主程式，所以這是一個相當簡單而易於明白的程式。第740行設定字形位置變數L與C的值為0，第750行讀入鍵盤被按字鍵的資料。

第760行的N＝ASC（A$）－31是要使下述指令FOR I=1 TO N能讀至該字鍵的資料，譬如“B”這個字的ASC函數值為66，所以N＝66－31＝35，由第100行算起的第35組DATA恰好是“B”。第770行將字形每次向右移動一個字形的位置。第780行以亂數產生器RND來指定字形的顏色。

第790行以FOR I＝0 TO 4來畫出每列的5個點，而這些點是由IF M THEN 來決定是否畫出，而FOR J=0 TO 6是把7列的點都畫出來。

另一個例子

在前面的程式裡，第760行每次執行時都要從頭把DATA讀入，而其實每次只用了最後一組的DATA，因此執行起來速度就慢了，而越是排在後頭的字形就越慢，譬如“Z”這個字就比“#”這個符號慢了。要解決這個問題，可以在INTEGER BASIC中使用GOTO ASC（AS）的方式叫程式去讀所按字鍵的相對應行號即可（此時每行行號要恰為其所代表字形的ASCII值）。但是在APPLESOFT的BASIC上，這個指令是不能使用的，GOTO之必須是數字才行（亦即是行號），不可以是變數，查看有關INTEGER BASIC與APPLESOFT的對照表，發現可以用ON ASC（A$）GOTO來代替，但是有一缺點是必須在GOTO以後將所有的行號列出，當目的地的數目太多的話，就顯得相當麻煩。因此，要想其他方式來解決。

GOTO……，READ這一組合指令的功能是可以由PEEK來取代的，而所要READ的DATA可以由POKE來存入記憶體，最大的一個困難是DATA欄中的數值是十進位的實數，而POKE I，J中的J是要存入記憶體的值，因此不可以超過255。如此要把程式中DATA的值直接POKE進入記憶位址，就要另想辦法來解決。

在底下的程式裡，第790行至第840行就是先把DATA的資料，利用其每一位數不是0就是1的性質，壓縮成爲二進位加權值，再POKE進入記憶位址內（注意，加權後的值已經比255小，但雖然是二進位加權，但是仍然以十進位被POKE）。當所有DATA均被POKE完後，由第20行的ONERR GOTO將程式轉移至第860行，繼續往下執行。

在第870行由鍵盤GET A$之後，就一連串的進行將該字鍵的資料以PEEK的方式由記憶位址中取出成爲D2，然後以第890至第900行來還原成原來的DATA形狀，再依次的畫在畫面上。

這樣修改後的程式，可以預見到的是，要先花一段時間來將全部DATA加以讀取並存入記憶體，但是以後就只對單一位址（連續7個位址）取出資料，而不必從頭開始了。在執行本程式時，須要等待約80秒鐘的時間，這是電腦在讀取所有資料，加以壓縮，並存入記憶位址的時間，在這段時間內，畫面爲黑色，且鍵盤的字鍵沒有作用。

程　式

```
]LIST

10   DEF FN M(A)= INT((A/X- INT(A/X))*X
     +0.05)
20   ONERR GOTO 860
100  DATA 0,0,0,0,0,0,0:REM
110  DATA 100,100,100,100,100,0,100:
     REM !
120  DATA 1010,1010,0,0,0,0,0: REM "
130  DATA 0,1010,11111,1010,11111,1010,
     ,0:REM #
140  DATA 100,11110,101,1110,10100,1111
     ,100:REM $
```

```
150 DATA 11 , 10011 , 1000 , 100 , 10 , 11001 ,
    11000 : REM %
160 DATA 10 , 101 , 101 , 10 , 10101 , 1001 , 10110
    : REM &
170 DATA 100 , 100 , 100 , 0 , 0 , 0 , 0 : REM ,
180 DATA 100 , 10 , 1 , 1 , 1 , 10 , 100 : REM (
190 DATA 100 , 1000 , 10000 , 10000 , 10000 ,
    1000 , 100 : REM )
200 DATA 100 , 10101 , 1110 , 100 , 1110 , 10101
    , 100 : REM *
210 DATA 0 , 100 , 100 , 11111 , 100 , 100 , 0
    : REM +
220 DATA 0 , 0 , 0 , 0 , 100 , 100 , 10 : REM ,
230 DATA 0 , 0 , 0 , 11111 , 0 , 0 , 0 : REM -
240 DATA 0 , 0 , 0 , 0 , 0 , 0 , 100 : REM ·
250 DATA 0 , 10000 , 1000 , 100 , 10 , 1 , 0 : REM /
260 DATA 1110 , 10001 , 11001 , 10101 , 10011
    , 10001 , 1110 : REM 0
270 DATA 100 , 110 , 100 , 100 , 100 , 100 , 1110
    : REM 1
280 DATA 1110 , 10001 , 10000 , 1100 , 10 , 1
    , 11111 ; REM 2
290 DATA 11111 , 10000 , 1000 , 1100 , 10000
    , 10001 , 1110 : REM 3
300 DATA 1000 , 1100 , 1010 , 1001 , 11111 , 1000
    , 1000 : REM 4
310 DATA 11111 , 1 , 1111 , 10000 , 10000 , 10001 ,
    , 1110 : REM 5
```

```
320 DATA 11100,10,1,1111,10001,10001,
    1110:REM 6
330 DATA 11111,10000,1000,100,100,100
    ,100:REM 7
340 DATA 1110,10001,10001,1110,10001
    ,10001,1110:REM 8
350 DATA 1110,10001,10001,11110,10000
    ,1000,111:REM 9
360 DATA 0,0,100,0,100,0,0:REM :
370 DATA 0,0,100,0,100,100,10:REM ;
380 DATA 10000,1000,100,10,100,1000,
    10000:REM <
390 DATA 0,0,11111,0,11111,0,0:REM =
400 DATA 10,100,1000,10000,1000,100,10
    :REM >
410 DATA 1110,10001,1000,100,100,0,100
    :REM ?
420 DATA 1110,10001,10101,11101,1101,1
    ,11110:REM @
430 DATA 100,1010,10001,10001,11111,
    10001,10001:REM A
440 DATA 1111,10001,10001,1111,10001,
    10001,1111:REM B
450 DATA 1110,10001,1,1,1,10001,1110:
    REM C
460 DATA 1111,10001,10001,10001,10001
    ,10001,1111:REM D
470 DATA 11111,1,1,1111,1,1,11111;REM E
```

```
480 DATA 11111,1,1,1111,1,1,1:REM F
490 DATA 11110,1,1,1,11001,10001,11110
    :REM G
500 DATA 10001,10001,10001,11111,10001
    ,10001,10001:REM H
510 DATA 1110,100,100,100,100,100,1110
    :REM I
520 DATA 10000,10000,10000,10000,10000
    ,10010,1100:REM J
530 DATA 10001,1001,101,11,101,1001,
    10001:REM K
540 DATA 1,1,1,1,1,1,11111:REM L
550 DATA 10001,11011,10101,10101,10001
    ,10001,10001:REM M
560 DATA 10001,10001,10011,10101,11001
    10001,10001:REM N
570 DATA 1110,10001,10001,10001,10001
    ,10001,1110:REM O
580 DATA 1111,10001,10001,1111,1,1,1
    :REM P
590 DATA 1110,10001,10001,10001,10101,
    1001,10110:REM Q
600 DATA 1111,10001,10001,1111,101,1001,
    10001:REM R
610 DATA 1110,10001,1,1110,10000,10001
    ,1110:REM S
620 DATA 11111,100,100,100,100,100,100
    :REM T
```

```
630 DATA 10001 , 10001 , 10001 , 10001 , 10001
    , 10001 , 1110 : REM  U
640 DATA 10001 , 10001 , 10001 , 10001 , 10001
    , 1010 , 100 : REM  V
650 DATA 10001 , 10001 , 10001 , 10101 , 10101
    , 11011 , 10001 : REM  W
660 DATA 10001 , 10001 , 1010 , 100 , 1010 ,
    10001 , 10001 : REM  X
670 DATA 10001 , 10001 , 1010 , 100 , 100 , 100
    , 100 : REM  Y
680 DATA 11111 , 10000 , 1000 , 100 , 10 , 1 ,
    11111 : REM  Z
690 DATA 11111 , 11000 , 11000 , 11000 , 11000
    , 11000 , 11111 : REM  ]
700 DATA 0 , 10000 , 1000 , 100 , 10 , 1 , 0 : REM
    BACK SLASH
710 DATA 11111 , 11 , 11 , 11 , 11 , 11 , 11111 :
    REM  [
720 DATA 0 , 0 , 100 , 1010 , 10001 , 0 , 0 : REM ∧
730 DATA 0 , 0 , 0 , 0 , 0 , 0 , 11111 : REM
    UNDERLINE
740 HOME : L = 0 : C = L : GR
770 AR = 16384
780 X = 10 : RESTORE
790 READ D2 : B = 1 : M = 0
800 FOR N=1 TO 5 : M=M+FN M( D2 )*B : D2 =
    INT( D2/X )
830 B = B * 2 : NEXT
```

```
840 POKE AR,M:AR=AR+1:GOTO 790
860 X=2
870 GET A$:IF A$=" "THEN TEXT:HOME:
    END
880 AD=7*(ASC(A$)-32)+16384
890 FOR I=0 TO 6:D2=PEEK(AD+I):M=0
    :B=1
900 FOR J=1 TO 5:M=M+FN M(D2)*B:D2=
    INT(D2/X):B=B*10:NEXT:P%(I)=M
    :NEXT
910 IF C<36 THEN 930
920 C=0:L=L+8:IF L>39 THEN L=0:C=L
930 R=INT(15* RND(1)+1)
940 FOR J=0 TO 6:FOR I=0 TO 4:A=P%
    (J)/10:P%(J)=A:M=A-P%(J):IF M
    THEN COLOR=R:PLOT C+I,L+J:GOTO
    960
950 COLOR=O:PLOT C+I,L+J
960 NEXT:NEXT:C=C+6:GOTO 870
970 END
```

程式三　潛水艇

這是一個完整的遊戲程式，結構複雜性屬於中等，很適合當做本書第二部份的例子來詳加說明。將程式鍵入後，畫面出現一段說明，本程式是戰艦與潛艇的遊戲，戰艦由遊戲者加以控制，當戰艦由畫面中央出現以後，就向右行駛，如果要令戰艦向左進行就按“ A ”鍵來控制，“ B ”鍵可以令戰艦停在某處投下深水炸彈。程式控制深水炸彈最多不超過十枚，潛艇最多不超過五艘，遊戲時間爲110秒鐘，如果時間用完時得分超過400分，則可再多出 80 秒鐘的時間，當船被潛艇的魚雷擊沈時，得分減半，而深水炸彈打到一艘潛艇時，依潛艇潛水的深淺而加分並發出三個響聲。當遊戲結束時，依所打下的潛艇數目再加上獎分。

程　式

```
]LIST

1  REM  ** DEPTH CHARGE
50  GOTO 5000
98  REM  **  DRAW SUBMARINE
100  REM
104  IF S(S) < 0 THEN  RETURN
105  HLIN SX(S) - 3,SX(S) + 3 AT SY
     (S) * 4: HLIN SX(S) - 3,SX(S) +
     3 AT SY(S) * 4 + 1
120  PLOT SX(S) + SDX(S),SY(S) * 4 -
```

```
    1
198 RETURN
199 REM  **  DRAW SHIP
200 FOR X = 3 TO 1 STEP  - 1: HLIN
    XSHIP - 7 + X,XSHIP + 7 - X AT
    5 + X: NEXT X
210 FOR X = 3 TO 5: HLIN XSHIP - 1
    ,XSHIP + 1 AT X: NEXT X: PLOT
    XSHIP,2
298 RETURN
299 REM  **  MOVE MINE
300 FOR M = 1 TO 10: IF D(M) < 0 THEN
    390
301 IF  SCRN( DX(M),DY(M)) = 0 THEN
    380
302 COLOR= 2: PLOT DX(M),DY(M):DY(
    M) = DY(M) + 2: IF DY(M) > 38 THEN
    370
305 IF  SCRN( DX(M),DY(M)) = 0 THEN
    380
310 COLOR= 8: PLOT DX(M),DY(M): NEXT
    M: RETURN
370 D(M) =  - 1: NEXT M: RETURN
380 FOR XX =  - 1 TO 1: COLOR= 1: HLIN
    DX(M) - 1,DX(M) + 1 AT DY(M) +
    XX: NEXT XX
381 FOR MM = 1 TO 3
382 POKE 789,30: POKE 790,20: CALL
    768
383 FOR MJ = 1 TO 10: NEXT MJ
384 NEXT MM
385 FOR XX =  - 1 TO 1: COLOR= 2: HLIN
```

```
     DX(M) - 1,DX(M) + 1 AT DY(M) +
     XX: NEXT XX
386 D =  INT ((DY(M) + 1) / 4): FOR
     XX = 1 TO 6: IF S(XX) > 0 AND
     SY(XX) = D THEN 388
387  NEXT XX: NEXT M: RETURN
388 S = XX: COLOR= 2: GOSUB 100:S(X
     X) =  - 1
389 SK = SK + 1:SCR% = SCR% + 4 * D
     : GOTO 370
390  REM
395  NEXT M: RETURN
398  REM  **  PRODUCE A NEW MINE
400 N = 0: FOR X = 1 TO 10: IF S(X)
      > 0 THEN N = N + 1
401  NEXT X: IF  INT ( RND (20 / N +
     1) * (20 / N + 1)) <  > 0 THEN
      RETURN
410  FOR X = 1 TO 10: IF M(X) < 0 THEN
     420
411  NEXT : RETURN
420 C = 1000: FOR Z = 1 TO 10: IF  ABS
     (SX(Z) - XSHIP) > C OR S(Z) <
     0 THEN 430
421 C1 = Z:C =  ABS (SX(Z) - XSHIP)

430  NEXT Z:M(X) = 1:MX(X) = SX(C1)
     :MY(X) = SY(C1) * 4 - 1: RETURN

598  REM  **  MOVE DEPTH CHARGE
600  FOR X = 1 TO 10: IF M(X) < 0 THEN
     690
```

```
601  COLOR= 2: PLOT MX(X),MY(X):MY(
     X) = MY(X) - 2: IF MY(X) > 8 THEN
     680
602  IF  SCRN( MX(X),MY(X)) = 15 THEN
     FLAG = 1
603 M(X) =  - 1
605  COLOR= 1: HLIN MX(X) - 1,MX(X)
      + 1 AT MY(X): HLIN MX(X) - 1,
     MX(X) + 1 AT MY(X) + 1: FOR ZZ
     Z = 1 TO 25
608  NEXT ZZZ: COLOR= 7: HLIN MX(X)
      - 1,MX(X) + 1 AT MY(X): HLIN
     MX(X) - 1,MX(X) + 1 AT MY(X) +
     1
610  GOTO 690
680  COLOR= 11: PLOT MX(X),MY(X)
690  NEXT X: RETURN
698  REM  **  THE SHIP HAVE BEEN SU
     NK!!
700  COLOR= 15
702  INVERSE : HOME : PRINT "YOU HA
     VE BEEN SUNK!!!": NORMAL
710  FOR X = 0 TO 10: FOR Y = 1 TO
     3: IF X + Y + 5 > 8 THEN 720
711  HLIN XSHIP - 7 + Y,XSHIP + 7 -
     Y AT 5 + X + Y
715  NEXT Y
720  FOR YY = 1 TO 3: IF X + YY + 3
      > 8 THEN 722
721  HLIN XSHIP - 1,XSHIP + 1 AT X +
     YY + 3: NEXT YY
722  FOR Z1 = 1 TO 30: NEXT Z1: FOR
     Z = 1 TO 8: COLOR= 7
```

```
725  HLIN 0,39 AT Z: NEXT Z: COLOR=
     15: NEXT X
726  FOR X = 9 TO 39: COLOR= 2: HLIN
     0,39 AT X: NEXT X: FOR X = 1 TO
     10:S = X: COLOR= 0: GOSUB 100:
     M(X) =  - 1:D(X) =  - 1: NEXT
     X
727  FOR X = 1 TO 200: NEXT X:XSHIP
      = 20: COLOR= 15: GOSUB 200
730 SCR% = SCR% / 2:FLAG = 0: RETURN

998  REM  **  TO MOVE THE SHIP
1000 W =  PEEK ( - 16384): POKE
     16368,0: IF W = 193 THEN 1010
1001  IF W = 194 THEN 1060
1002  GOTO 1020
1010 XN = XSHIP - 3: GOTO 1030
1020 XN = XSHIP + 3
1030  IF XN < 8 OR XN > 32 THEN 105
     0
1040  COLOR= 7: GOSUB 200:XSHIP = X
     N: COLOR= 15: GOSUB 200
1050  REM
1060  FOR X = 1 TO 10: IF D(X) < 0 THEN
     1100
1061  NEXT X: RETURN
1100 D(X) = 1:DX(X) = XSHIP:DY(X) =
     9: RETURN
1998  REM  **  TO PRODUCE A NEW SUB
     MARINE
2000  FOR X = 1 TO 5: IF S(X) < 0 THEN
     2010
```

```
2001  NEXT X:SS =  - 1: RETURN
2010 D =  INT ( RND (6) * 6 + 4): FOR
     Y = 1 TO 5: IF S(Y) > 0 AND SY
     (Y) = D THEN 2010
2011  NEXT Y
2020 S(X) = 1:SY(X) = D
2030 SDX =  INT ( RND (2) * 2):SX(X
     ) = 4 + SDX * 31:SDX(X) =  - 1
     : IF SDX = 0 THEN SDX(X) = 1
2035 SS = X
2040  RETURN
2500  REM  **  TO MOVE OR DESTROY A
      SUBMARINE
2510 NUMBER = 0: FOR S = 1 TO 10: IF
     S(S) < 0 THEN 2700
2511  COLOR= 2: GOSUB 100:SX(S) = S
     X(S) + SDX(S)
2520  IF SX(S) < 4 OR SX(S) > 35 THEN
     2600
2521 NUMBER = NUMBER + 1: COLOR= 0:
      GOSUB 100: NEXT S
2550  REM :
2560  IF  INT ( RND (NUMBER + 1) *
     (NUMBER + 1)) = 0 THEN  GOSUB
     2000
2561  IF SS < 0 THEN  RETURN
2562 S = SS: COLOR= 0: GOSUB 100: RETURN
2600 S(S) =  - 1: NEXT S: GOTO 2550

2700  REM
2710  NEXT S: GOTO 2550
3000  REM  **  TO BEGINE A NEW GAME
```

```
3005  GR : CALL  - 936
3010  FOR X = 0 TO 39: COLOR= 7: IF
     X > 8 THEN  COLOR= 2
3011  HLIN 0,39 AT X: NEXT X
3020  FOR X = 1 TO 10:M(X) =  - 1:S
     (X) =  - 1:D(X) =  - 1: NEXT X
     :XSHIP = 20: COLOR= 15: GOSUB
     200: GOSUB 2000: COLOR= 0:S =
     1: GOSUB 100
3030  REM
3498  RETURN
3499  REM  **  INTRODUCE
3500  INVERSE : PRINT "           D
     E P T H  C H A R G E        ": PRINT
     "  YOU ARE THE CAPTIAN OF WORL
     D WAR  "
3510  PRINT "II DESTROYER. TO LAUNC
     H A DEPTH CHARGE": PRINT "PRES
     S KEY'A'. IF THE       "
3520  PRINT "CHARGE HITS A SUBMARIN
     E, YOU WILL GET ": PRINT "A CE
     RTAIN NUMBER OF POINTS, ACCORD
     ING "
3530  PRINT "TO HOW DEEP THE SUB WA
     S.                ": PRINT "  AS
     THE SUBMARINES PASS BENETH YOU
        "
3540  PRINT "THEY MAY RELEASE MINES
      WHICH MOVE      ": PRINT "SLOW
     LY UPWARD. IF YOU LET A MINE H
     IT  "
3550  PRINT "YOUR SHIP, YOU WILL LO
```

```
SE 1/2 OF YOUR  ": PRINT "POIN
TS. TO STEER AROUND THE MINES,
 USE"
3560  PRINT "KEY'B'.
                              ": PRINT
      "WHEN TIME          REACHES ZER
      O THE GAME ENDS    "
3570  PRINT "AND YOU WILL BE AWARDE
      D 20 POINTS FOR ": PRINT "EACH
       SUB YOU SANK.
          "
3580  PRINT "       GOOD LUCK CAPTAI
      N !!!!            ": NORMAL : INPUT
      "HIT RETURN WHEN READY";A$: RETURN

4000  REM  **  INITIALIZE
4040  DIM D(10),DX(10),DY(10),M(10)
      ,MX(10),MY(10)
4042  DIM S(10),SDX(10)
4045  DIM SX(10),SY(10)
4050 SCR% = 0:SK = 0
4055  TEXT : HOME
4900 TIME = 110
4910 F1 = 0
4999  RETURN
5000  REM  ----------------
5010  REM !  MAIN PROGRAM  !
5020  REM  ----------------
5022  FOR I = 0 TO 20: READ W: POKE
      768 + I,W: NEXT
5024  DATA   173,48,192,136,208,5,2
      06,22,3,240,9,202,208,245,174,
```

```
,21,3,76,0,3,96
5030  GOSUB 4000
5040  GOSUB 3500
5050  GOSUB 3000
5060  GOSUB 400: GOSUB 300
5070  GOSUB 1000
5075  IF FLAG = 1 THEN 5100
5080  GOSUB 600
5090  GOSUB 2500
5095  POKE 789,70: POKE 790,40:A =
      INT (TIME / 2) * 2: IF TIME -
     A = 1 THEN  POKE 790,10
5096  CALL 768
5100  IF FLAG = 1 THEN  GOSUB 700
5105  HOME
5110 TIME = TIME - 1: PRINT "SUBS S
     UNK=";SK,"SCORE=";SCR%,"TIME="
     ;TIME;"     "
5120  IF TIME > 0 THEN 5060
5122  IF SCR% < 400 OR F1 = 1 THEN
     5125
5123 F1 = 1:TIME = 80: HOME
5124  CALL  - 384: PRINT "      EXTE
     NDED PLAY !!!": NORMAL : GOTO
     5060
5125  IF SK = 0 THEN  END
5130  FOR X = 1 TO SK: POKE 789,10:
      POKE 790,20: CALL 768:B = B +
     20: VTAB 22: PRINT "BONUS=";B
5140  NEXT X: VTAB (22): HTAB (13):
      CALL  - 384: PRINT "TOTAL (SC
     ORE+BONUS)=";SCR% + B;: CALL  -
     380: END
```

程式說明

表 2-2 副程式列表

副　　程　　式	行　號
畫潛艇	100
畫戰艦	200
移動深水炸彈（戰艦）	300
發射魚雷	400
移動魚雷（潛艇）	600
戰艦下沈	700
移動戰艦及投深水炸彈	1000
產生新潛水艇	2000
移動潛水艇	2500
起始畫面	3000
遊戲說明	3500
初值設定	4000

主程式由5000開始，首先在5022讀入發聲副程式，然後至4000，3500，3000等副程式設定初值及畫面，同時印出遊戲說明，當遊戲者瞭解說明事項之後，由3580行的 INPUT A$ 回到主程式來。

5060行的GOSUB 400及GOSUB 300就開始了程式的運轉，戰艦的移動由SUB 1000來控制，如果船被魚雷打中，則FLAG＝1並到SUBROUTINE 700使戰艦沈下，再回到5110行來計算積分、時間，如果時間用完，再判斷是否增加時間，否則加上獎分而結束。若FLAG≠1，則計算時間是否結束，若是則進入上述的過程，若否則由5120行跳至5000行繼續進行遊戲，其流程見圖2.7。

本程式較複雜之處在於其由數個副程式構成，而且有些情況副程式之間亦相關聯，因此欲詳細瞭解則必對各個副程式個別加以研究。其中比較簡單的幾個副程式是第100、第200、第3000、第3500及第4000等副程式，第4000只是設定初值，第3500則在螢幕上印出程式說明，而

第3000副程式則畫出最初的畫面及將M（X），S(X)，D(X)等陣列變數設定爲－1，以上三個變數是分別用來判別是否存在一個魚雷、一艘潛艇或一顆深水炸彈，其值若爲－1則不存在，若大於0則存在。

第300副程式是以FOR M＝1 TO 10的迴路來檢查是否畫面存在深水炸彈，若找到第一顆存在的深水炸彈（由D(1)找起），則由301行的SCRN（DX(M)，DY(M)）＝0來查看是否碰上潛艇了（潛艇爲黑色COLOR＝0），若不是則由342行及310行來使深水炸彈下沈兩格位置，如果此時炸彈已沈到底，則至370行將此炸彈註銷（即D(M)＝－1）。若是潛艇被打中，則至380行畫出紅色的火焰，並由381行至384行呼叫發聲副程式（CALL 768），發出三個“嘀”的響聲，然後消去火焰圖形，再註銷潛艇（S（XX)＝－1），然後再消去潛艇圖形（潛艇之畫出及消去是經由100副程式執行的）。

第400副程式是以FOR X＝1 TO 10的迴路來找出是否存在有潛艇（S（X）＞0），若有則將該潛艇號碼記爲N，接着以亂數產生器RND來決定是否令該潛艇發射一枚魚雷，亂數值是設定成只有0與1，若1則回到主程式，若0則至410的FOR X＝1 TO 10來找出小於0的M（X），若M（X）大於0表示已存在三枚魚雷，所以繼續找小於0的M（X），若M（X）＜0則至420行來判斷，找出與戰艦水平距離最近的潛艇令其發射魚雷。

第600副程式是以FOR X＝1 TO 10的迴路來找出是否存在魚雷，若M（X）＜0則至690行繼續找下一個，若找到則先將它由畫面消去，並令其座標上升兩格，並查看是否碰上戰艦或已浮上水面，若是則畫出火焰並註銷該魚雷，若船被擊到則FLAG＝1。若尚未到水面上，則至680將魚雷畫出（注意此時已浮上兩格了）。

第700副程式是當主程式發現FLAG＝1時，由5100行轉移過來的，首先由702行在螢幕以反白字體（INVERSE）印出“你已被擊沈”的字幕，接着由710行至721行畫出船漸漸下沈的圖形，再由722行至727行來清除畫面，若潛艇存在則保留之，但深水炸彈及魚雷則註銷並從畫面

清除。730行則將已累積分數除半，FLAG歸零。

第1000副程式是以PEEK(－16384)來檢知遊戲者是否按下"A"鍵或是"B"鍵，若A則戰艦左移三格，若B則原地投彈，若不按鍵，則向右移動，船的移動是以200副程式來畫出新位置並消去舊位置。投彈是以1060行及1100行來令D(X)＝1以產生新的深水炸彈。及設定位置爲DX(X)＝XSHIP及DY(X)＝9，如此當回到主程式再轉到300副程式時就可投下一顆炸彈了。

第2000副程式是以FOR X＝1 TO 5的廻路來找出是否存在潛艇(最多5艘)，若否則以RND亂數產生器來決定新潛艇的深度並判斷是否與原先潛艇重疊同一航線(即水深)，接着由2020行來產生一新潛艇(S(X)＝1)，至於2030行則是以RND來任意指定潛艇是由螢幕左邊或右邊出現的，若是右邊則向左移動，若是左邊則向右移動。

第2500副程式是以FOR S＝1 TO 10來找出是否已有潛艇產生，若無則以RND亂數來決定是否至第2000副程式產生新潛艇。若已有潛艇，則將其原位置之圖形消去，再判斷是否超過畫面，若是則註銷，若否則座標移動一格並在新座標畫出新圖形來。

由於本程式是一次處理三種陣列變數，因此要處理的變數達數十個之多，並且要做各種判斷，因此若要完全瞭解，只有由各副程式分別研究，然後再整體組合起來才能有一完整的概念，程式中對多變數的判斷處理方式可供學習之處很多，希望藉由研習此程式會提供讀者一些有用的經驗。

流程圖

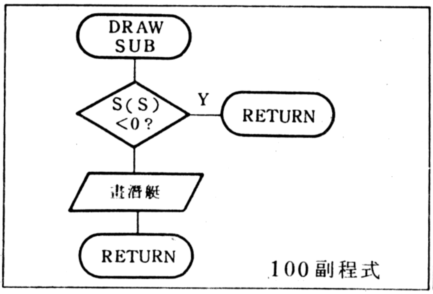


圖 2-7 (a)

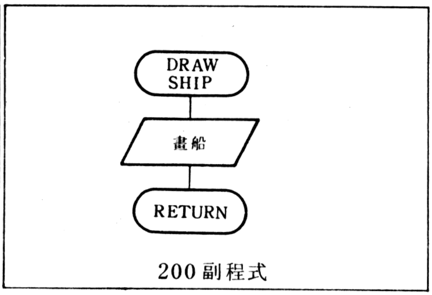
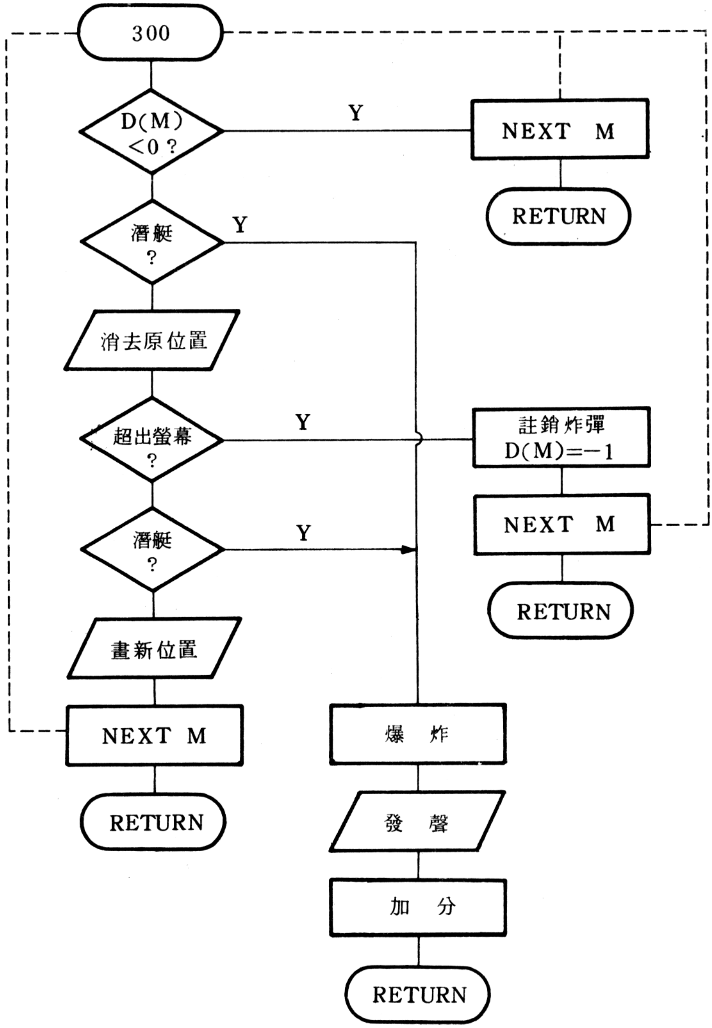


圖 2-7 (b)


300
D(M)
<0 ?
Y
NEXT M
RETURN
潛艇
?
Y
消去原位置
超出螢幕
?
Y
註銷炸彈
D(M)=-1
NEXT M
RETURN
潛艇
?
Y
畫新位置
NEXT M
RETURN
爆 炸
發 聲
加 分
RETURN

300副程式

圖 2-7(c)

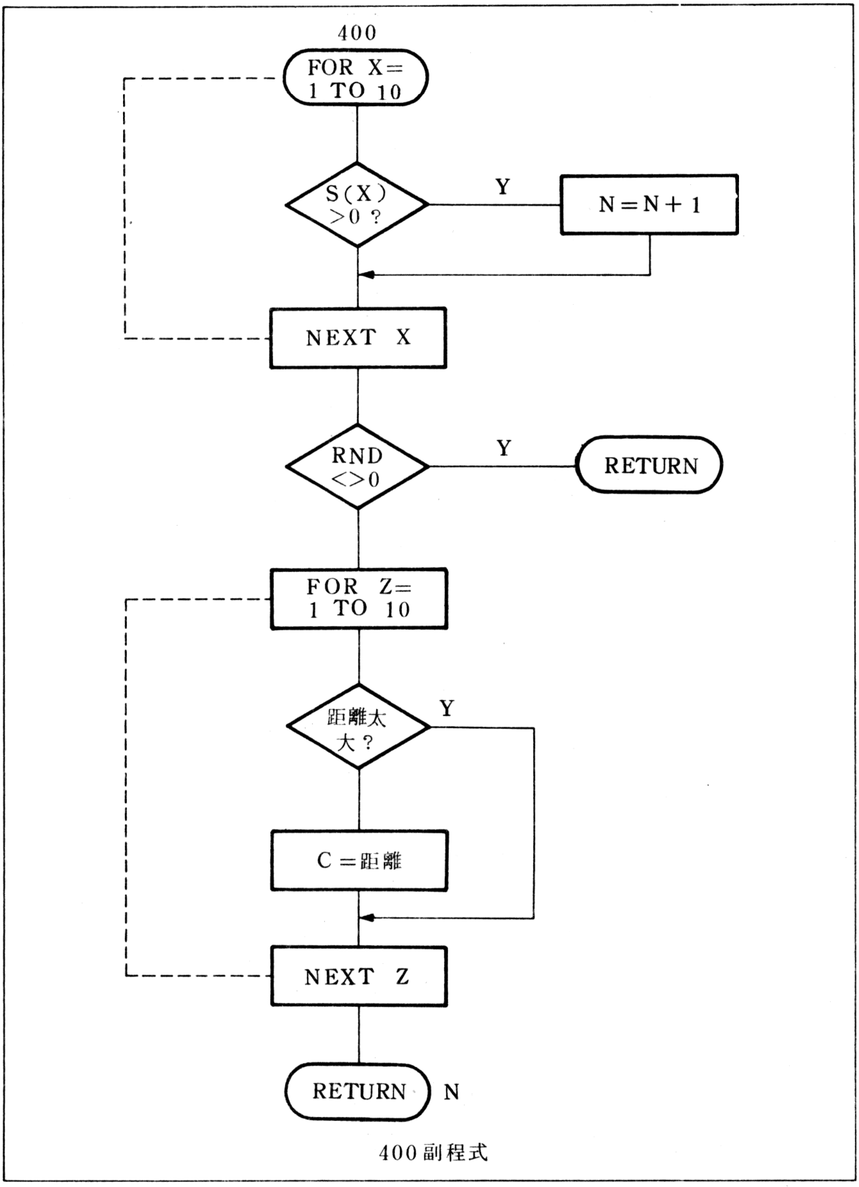
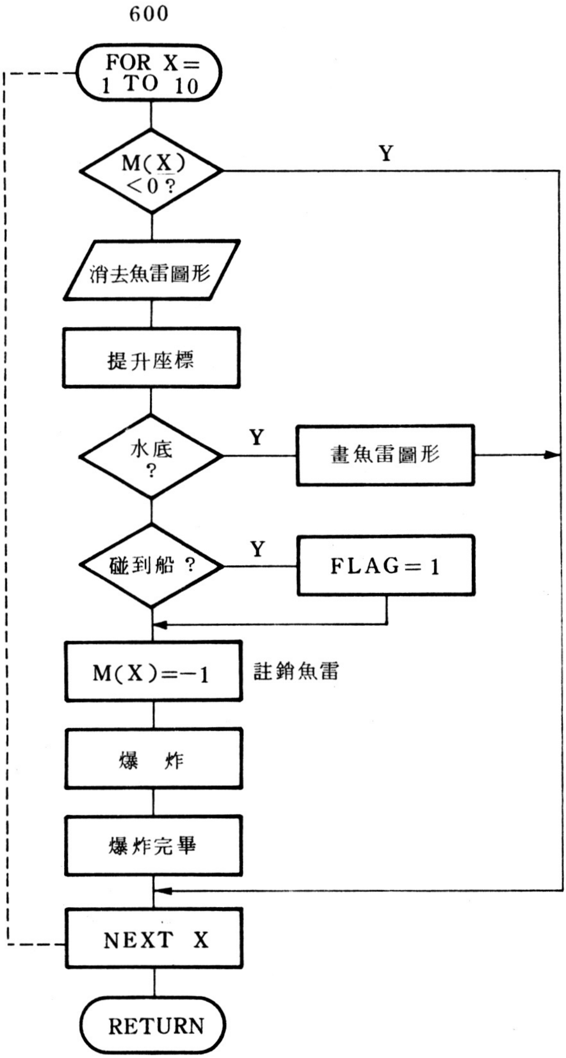

400
FOR X= 1 TO 10
S(X) >0 ?
Y
N=N+1
NEXT X
RND <>0
Y
RETURN
FOR Z= 1 TO 10
距離太大？
Y
C =距離
NEXT Z
RETURN
N
400副程式

圖 2-7 (d)


600
FOR X=
1 TO 10
M(X)
<0?
Y
消去魚雷圖形
提升座標
水底
?
Y
畫魚雷圖形
碰到船?
Y
FLAG=1
M(X)=-1
註銷魚雷
爆　炸
爆炸完畢
NEXT X
RETURN

600副程式

圖 2-7(e)

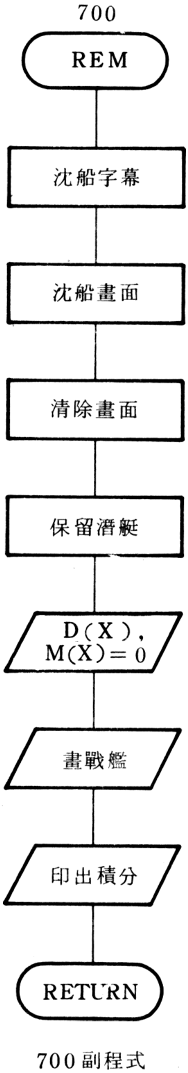

700
REM
沈船字幕
沈船畫面
清除畫面
保留潛艇
D(X), M(X)=0
畫戰艦
印出積分
RETURN
700副程式

圖 2-7(f)

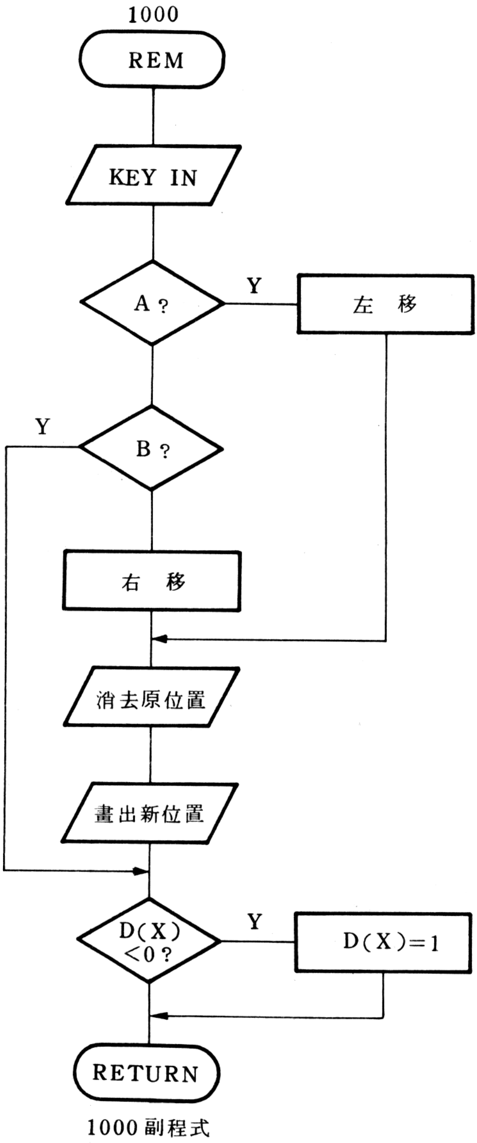
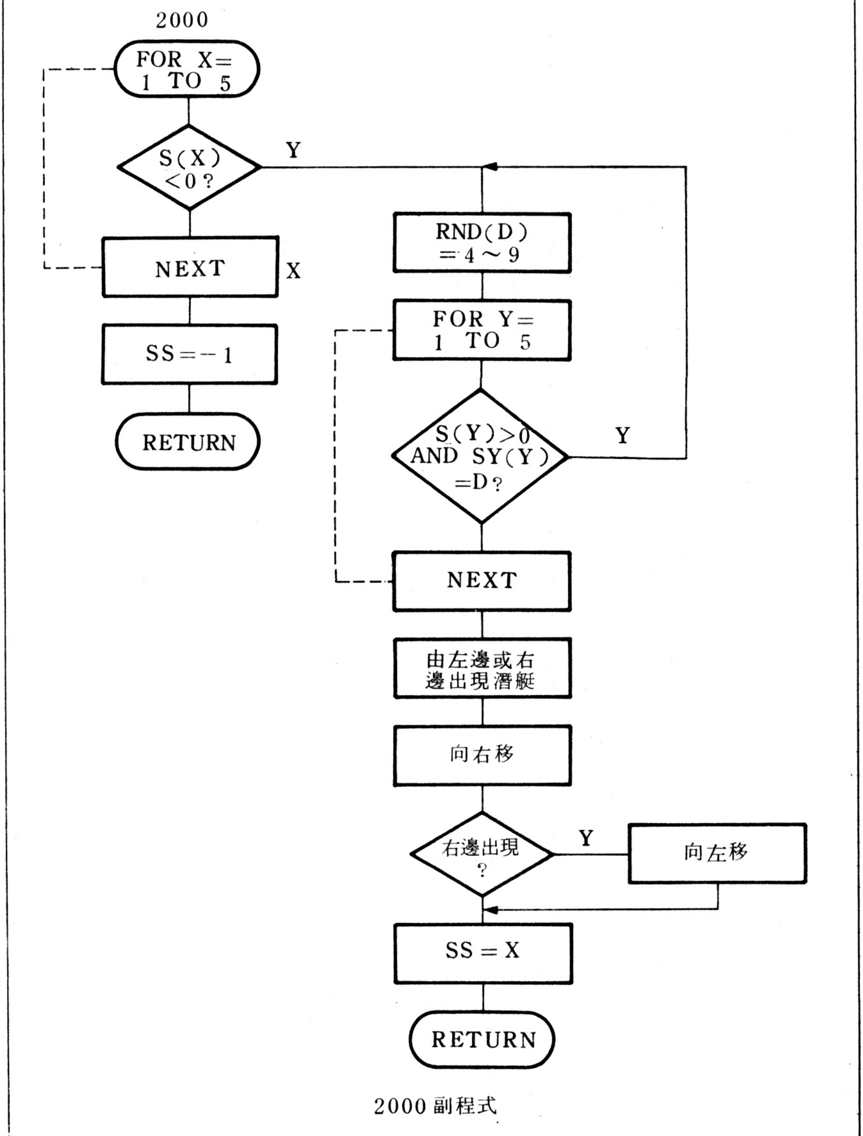
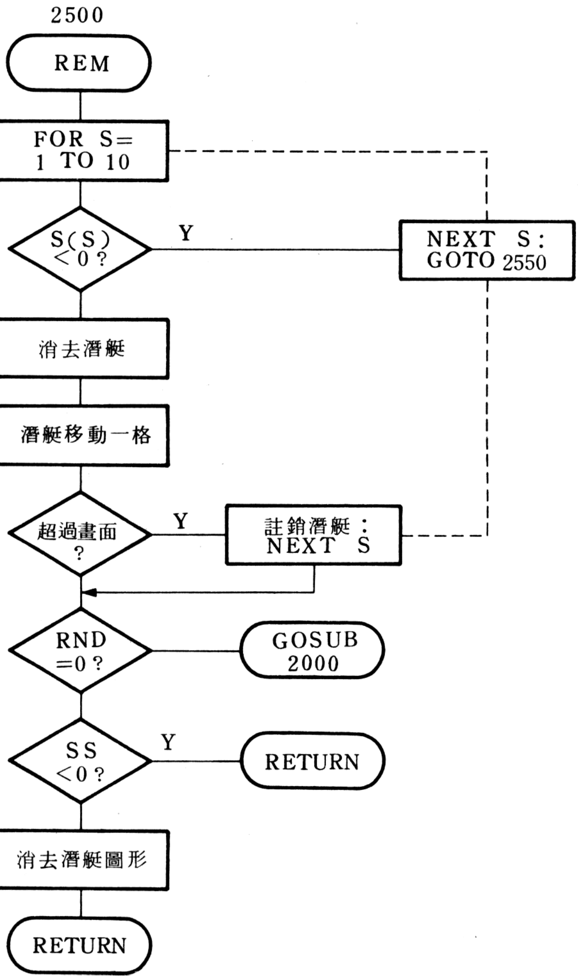

1000
REM
KEY IN
A ?
Y
左 移
B ?
Y
右 移
消去原位置
畫出新位置
D(X) <0 ?
Y
D(X)=1
RETURN
1000副程式

圖 2-7(g)



2000 副程式

圖 2-7(h)



2500副程式

圖 2-7(i)

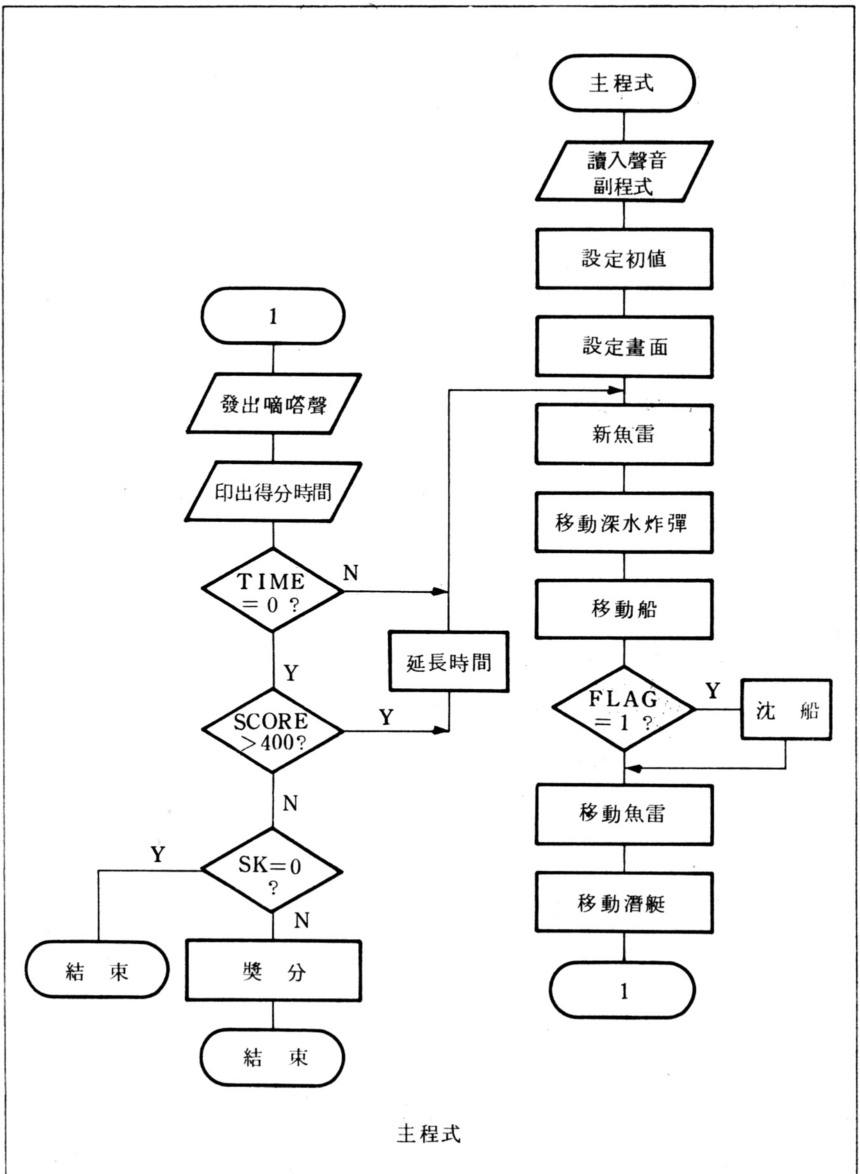

主程式
讀入聲音
副程式
設定初值
設定畫面
新魚雷
移動深水炸彈
移動船
FLAG
=1 ?
Y
沈　船
移動魚雷
移動潛艇
1
1
發出嘀嗒聲
印出得分時間
TIME
=0 ?
N
Y
延長時間
SCORE
>400?
Y
N
SK=0
?
Y
N
結　束
獎　分
結　束
主程式

圖 2-7(j)

程式四 賽馬

這是一個簡單的遊戲程式，以低解像畫面的圖形來模仿賽馬的情形。除了當成一個遊戲之外，更重要的是可藉由對本程式的分析而探究一般遊戲程式的結構。因爲本程式相當簡短，很容易明白其流程，是很適合於初學的一個實例。

程 式

```
]LIST

10  TEXT : HOME
11  INVERSE : PRINT  TAB( 16)"HORSE
      RACE                   ": NORMAL

15  DIM N$(27)
20  FOR I = 0 TO 27: READ N$(I): NEXT

25  PRINT : PRINT : PRINT : PRINT
30  PRINT  TAB( 8)"WELLCOME TO VIDE
     O DOWNS !"
40  PRINT : PRINT : PRINT
50  PRINT : PRINT  TAB( 12): INPUT
     "HOW MANY PLAYERS ";W
80  PRINT
```

```
90  FOR Y = 1 TO W: PRINT : PRINT  TAB(
     12)"NAME OF PLAYER ";Y;" ";: INPUT
     B$(Y): NEXT
110  FOR U = 1 TO W:B(U) = 50000: NEXT
130  FOR L = 1 TO 5:S =  INT (30 *
      RND (1)) + 1:R(L) = S: NEXT
150  HOME
170  INVERSE : PRINT "P A Y O F F S
      F O R H O R S E S": NORMAL
190  FOR Y = 1 TO 5
195 N =  INT ( RND (1) * 28):SW = 0

200  FOR J = 1 TO Y - 1: IF N(J) =
     N THEN SW = 1
205  NEXT J: IF SW = 1 THEN 195
210  PRINT : HTAB 3
220 N(Y) = N
230  PRINT Y;")";N$(N(Y));"PAYS";R(
     Y);"TO 1": NEXT Y
250  FOR Y = 1 TO W: PRINT : VTAB 1
     4: PRINT B$(Y);" ";"HAS $";B(Y
     )
270  PRINT : INPUT "HOW MUCH DO YOU
      WANT TO BET?";P(Y): IF P(Y) <
      = B(Y) THEN 290
280  PRINT "COME ON, YOU DON'T HAVE
      THAT MUCH .": GOTO 270
290  PRINT : INPUT "WHICH NUMBER HO
     RSE?";Z(Y): IF Z(Y) > 5 THEN 2
     90
310  FOR Q1 = 14 TO 21
330  VTAB Q1: PRINT "
```

```
     "
350  NEXT : NEXT
370  HOME : GR
390  COLOR= 12: FOR I = 0 TO 39: HLIN
     0,39 AT I: NEXT I
410 A = 1:X(1) = 2:Y(1) = 2:C(1) =
     2: COLOR= 2: GOSUB 550
430 A = 2:X(2) = 2:Y(2) = 10: COLOR=
     7:C(2) = 7: GOSUB 550
450 A = 3:X(3) = 2:Y(3) = 18: COLOR=
     8:C(3) = 8: GOSUB 550
470 A = 4:X(4) = 2:Y(4) = 26: COLOR=
     13:C(4) = 13: GOSUB 550
490 A = 5:X(5) = 2:Y(5) = 34: COLOR=
     1:C(5) = 1: GOSUB 550
500  FOR S = 1 TO 50:A =  PEEK ( -
     16336): NEXT S
510 A =  INT (5 *  RND (1)) + 1: GOSUB
     630
530  GOTO 510
550  PLOT X(A) + 1,Y(A) - 1: PLOT X
     (A) + 2,Y(A) - 1: PLOT X(A) -
     2,Y(A): PLOT X(A) - 1,Y(A): PLOT
     X(A),Y(A): PLOT X(A) + 1,Y(A)
570  PLOT X(A) - 2,Y(A) + 1: PLOT X
     (A) - 1,Y(A) + 1: PLOT X(A),Y(
     A) + 1: PLOT X(A) + 1,Y(A) + 1

590  PLOT X(A) - 2,Y(A) + 2: PLOT X
     (A) - 2,Y(A) + 3: PLOT X(A) +
     1,Y(A) + 2: PLOT X(A) + 1,Y(A)
      + 3
```

```
600  COLOR= 15: VLIN 0,39 AT 39
610  RETURN
630  COLOR= 12: GOSUB 550:B =  INT
     ( RND (1) * 3 / R(A) ^ .25) +
     1:X(A) = X(A) + B: IF X(A) <  =
     37 THEN  COLOR= C(A): GOSUB 55
     0: RETURN
650 T = A
660  FOR S = 1 TO 150:A =  PEEK ( -
     16336): NEXT S
670  GOTO 930
930  TEXT : HOME : INVERSE : PRINT
     "***** WINNINGS AND LOSINGS **
     ***": NORMAL
950  PRINT : PRINT "THE WINNER IS N
     UMBER";T;",";N$(N(T))
970  FOR Y = 1 TO W: PRINT : IF T =
     Z(Y) THEN L = T: PRINT B$(Y);"
     HAS WON $";P(Y) * R(L):B(Y) =
     B(Y) + (P(Y) * (R(L) - 1)): GOTO
     1010
990  PRINT B$(Y);"HAS LOST $";P(Y):
     B(Y) = B(Y) - P(Y)
1010  NEXT : PRINT : PRINT "ANOTHER
      RACE (Y/N)";: INPUT F$: IF F$
      = "Y" THEN 130: END
2000  DATA  "LIGHTNING FLASH","DADD
     Y'S LITTLE FILLY","GLUE FACTOR
     Y","DOG CHOW","CHECKERS","HONE
     YT BEE"
2010  DATA  "FOOT LOCKER","PATTY ME
     LT","MEADOW MUFFIN","BUFFALO C
```

```
        HIPS","ALL AGLOW","MERCHANT MA
        RINE"
2020  DATA  "IGLOO","ICE PACK","ROC
        KTY ROAD","HAYWIRE","NASHUA PR
        IDE","HOME STRETCH","SANDLEFOO
        T","ONION BREATH"
2030  DATA  "SADDLE SORES","MUSTARD
         PLASTER","BARNSTORMER","GRAIN
         BELLY","HOOF'N MOUTH","BONEY
        BACK","CHUBBY RUMP","SIDE SADD
        LE"
```

程式說明

程式分成兩段，第370行以前是由電腦與參加遊戲者交談，第370行以後是畫馬的程式以及勝負的比賽。

首先由第 10 行的 HOME 指令來清除畫面後，由第 11 行印出反白的“賽馬”字樣，第 20 行讀入 28 匹馬的名字。第 50 行詢問參加的人數“W”，並由第 90 行來讀入參加者的姓名“B$(Y)”。在經過110行及130行來設定參加者最初每人所擁有的點數或金額B(U)=50000，以及以S的亂數來指定每匹馬得獎的金額R(L)。第190行至第230行用來由 28 匹賽馬中選出任意5匹並指定其得獎金額，其中第200至205行是防止重覆選取的發生。第250行至第350行用以詢問參加者對於所列出的5匹馬選出一匹並下注金額。第270行中的 IF 是防止下注者的金額超過其所剩的金額。

第370行將畫面清除並指定低解像畫面。第390行將畫面的底色設爲綠色(COLOR= 12)，接着由410至第490行來定出5匹馬在畫面上的位置，其中X之初值均爲2，而Y值則爲各跑道的分配，C(1)、C(2)、……等是用來代替COLOR做爲變數用(因爲COLOR爲APPLE Ⅱ的內含字 RESERVE WORD，不可當做變數名)，第550行至610

行的副程式（SUB ROUTINE）是在X(A)及Y(A)所指定的座標位置畫出馬來。第500行是發出賽馬聲音的程式。第510行及530行是任意指定一匹馬並以630行的副程式來令其前進，在第630行，COLOR = 12 : GOSUB 550是將馬以底色來畫出，亦即消去馬的形象，並以B的亂數使X(A)=X(A)+B在新位置畫出馬來，注意此處B= INT(RND(1)＊3/R(A) .25)的分母是用R(A)，所以獎金多的馬(R(A)的值大)所前進的速度就慢了，所以大家不可儘挑獎金倍數大的選。

在APPLESOFT的指令中，IF-THEN是比較特別的，當IF的條件不成立時，所有在其後邊的指令，甚至以“：”分隔的其他敍述也一併被忽略而直接進行至下一行號。所以在第630行的IF X(A) < 37不成立時，亦即該馬已到達終點，則程式就執行到第650行指定T = A爲得獎者，第660行發出勝利的聲音。第930行至第990行印出參加者所剩餘的金額或點數，第1010行詢問是否要做另一次競賽，第2000行以後至2030行則爲28匹馬的名字資料（DATA）。

程式改進

在執行本程式時，發現畫馬的速度太慢，因此畫面有些不自然，如果以監督程式（MONITOR）的副程式來執行550行及630行的兩個畫圖副程式，則速度當可改善。另外，賽馬的過程稍嫌單調，如果在中途設立障礙物或以1-3節的副程式來改進發聲方法，當可令程式更多彩多姿了。

流程圖

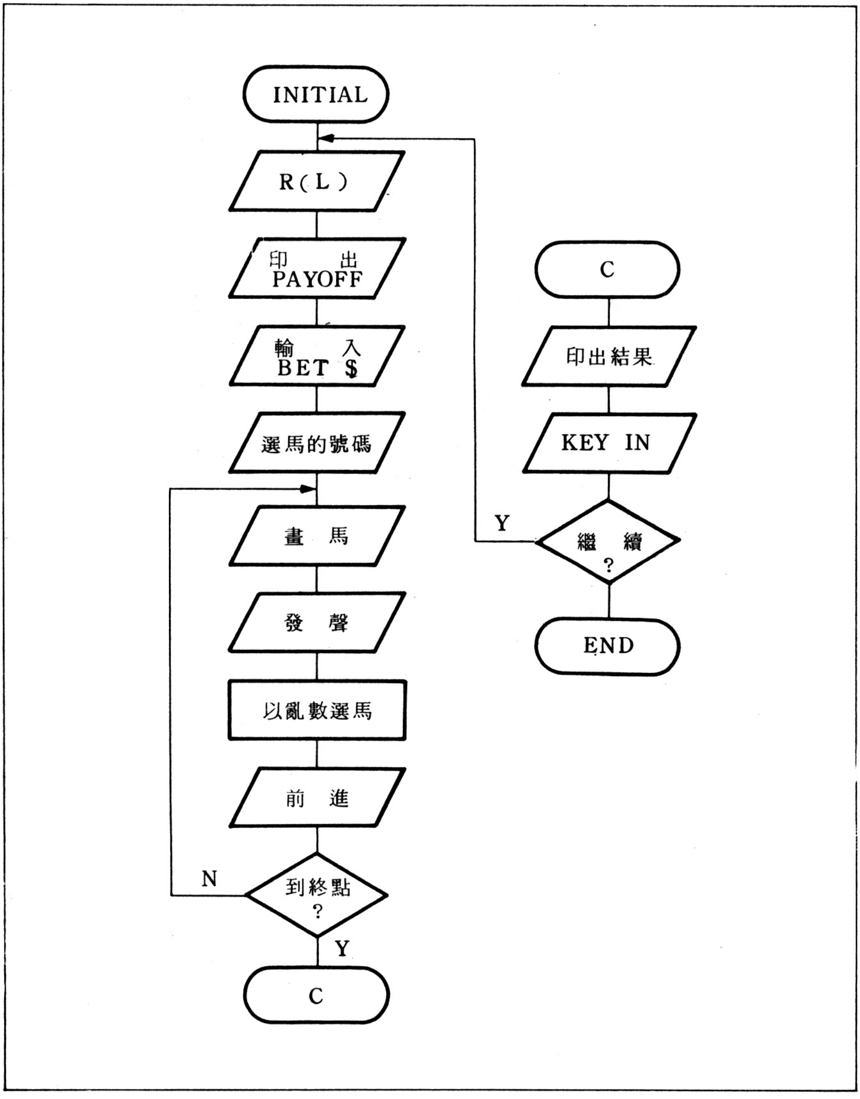
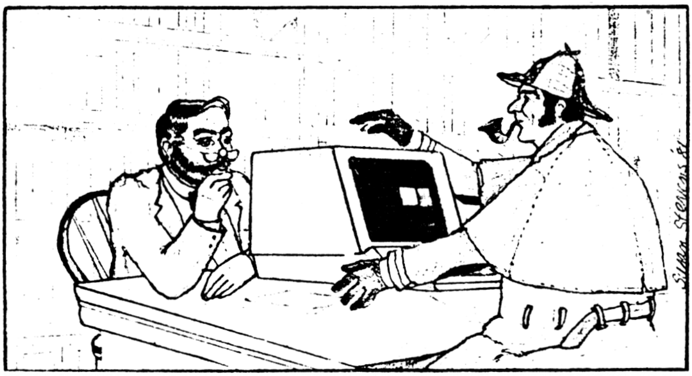

INITIAL
R(L)
印 出
PAYOFF
輸 入
BET $
選馬的號碼
畫 馬
發 聲
以亂數選馬
前 進
到終點
?
N
Y
C
C
印出結果
KEY IN
Y
繼 續
?
END

圖 2-8

程式五　顏色的密碼

習稱爲“主人的智力（MASTER－MIND）”的一種遊戲，通常由兩個人交換出題由對方猜出答案，出題的人在紙上排下四種顏色，但不令對方知道，接着由猜題人寫下四個顏色，這時出題人就根據其答案指出有幾色是顏色答對但位置排錯，有幾色是顏色對而位置也對，猜題人再據此分析判斷繼續作答而終至找出答案。這種遊戲必須集中注意力，愼密思考才會有好成績，因此做爲小孩子益智、邏輯思考的訓練是非常適當的，即使是對成人而言也是富有挑戰性的，而利用電腦來玩這種遊戲更能增加其多樣化的興趣吧！



程 式

```
]LIST

1  REM  COLOR FOR APPLF
5  DIM X$(12),CP(12),CC(12)
6  GOTO 100
10  REM  **DELAY LOOP SUBROUTINE**
15  FOR I = 1 TO 5000: NEXT : RETURN

20  REM  **FLASHING COLOR**
25 ZC =  SCRN( H,2 * V)
26  IF ZB = 5 THEN ZB = ZC: GOTO 30

27 ZB = 5
30  COLOR= ZB: PLOT H,2 * V
35  FOR I = 1 TO 15:A = 0:A =  PEEK
     ( - 16384): POKE  - 16368,0
40  IF A < 128 THEN  GOTO 50
43 A$ =  CHR$ (A - 128):I = 15:F1 =
     1: GOTO 50
50  NEXT I
55  IF F1 <  > 1 THEN 26
56 F1 = 0: RETURN
84  REM
85  REM  ** OPTION LIST ROUTINE **
86  REM
90  HTAB 1: VTAB 23: PRINT "PRESS<X
     > TO CHANGE GUESS;<E> TOGRADE;
     ";
91  HTAB 10: VTAB 24: PRINT "PRESS<
     S> TO SWITCH MODES.";
```

```
92  GET A$: RETURN
94  REM
95  REM  ** MAIN PROGRAM LISTING **

96  REM
100  TEXT : HOME : VTAB 10: PRINT "
     **  COLOR  **": PRINT "**  COD
     E   **": PRINT "**  COMBO  **"
     : PRINT
101  GOSUB 15
103  FOR I = 1 TO 12:X$(I) = "":CP(
     I) = 0:CC(I) = 0: NEXT
105  PRINT : PRINT : PRINT "HOW MAN
     Y COLORS DO YOU": INPUT "WANT
     TO WORKWITH (3-8)  ";C
110  IF C < 3 OR C > 8 THEN 100
115  PRINT : INPUT "HOW LARGE A TAB
     LEAU (3-8)  ";T
120  IF T < 3 OR T > 8 THEN 100
125  PRINT : INPUT "HOW MANY GUESS
     WILL YOU NEED  (4-12)";G
130  IF G < 4 OR G > 12 THEN 100
131  PRINT : PRINT "WHITCH MODE, ";
     : POKE 50,127: PRINT "C";: POKE
     50,255: PRINT "OLOR OR";: POKE
     50,127: PRINT "A";: POKE 50,25
     5: PRINT "LPHABET?";: GET A$
132 S = 1: IF A$ = "C" THEN S = 2: GOTO
     140
133  IF A$ <  > "A" THEN 131
134  REM
135  REM  ** RANDOM PROBLEM GENERAT
```

```
      OR **
136   REM
140   DATA  "R","B","G","Y","O","W",
      "P","A"
141   REM
145   FOR I = 1 TO 8: READ A$(I): NEXT

150   RESTORE
155 R$ = "": FOR I = 1 TO T:R$ = R$
       + A$( INT ( RND (1) * C + 1))
      : NEXT
156   REM
160   REM  ** SET UP DISPLAY **
161   REM
165   IF S = 1 THEN  GOSUB 600: GOTO
      185
166   GOSUB 600: GOSUB 400
176   REM
180   REM  ** OBTAIN PLAYERS GUESS *
      *
181   REM
185   FOR K = 1 TO G
190 G$ = ""
195   FOR L = 1 TO T
196 ZZ = 120 + 40 * K + 2 * L:V =  INT
      (ZZ / 40):H =  INT (ZZ - 40 *
      V)
200   IF S = 1 THEN  HTAB H: VTAB V:
       GET A$: PRINT A$;: GOTO 210
201   IF S = 2 THEN ZC = 0: HTAB 39:
       VTAB 24:X = H:Y = 2 * V: GOSUB
      25: GOSUB 500
```

```
210 G$ = G$ + A$
211 X$(K) = G$
213 A$ = "":A = 0
215  NEXT L
220  GOSUB 90
221  IF A$ = "S" AND S = 1 THEN  GOSUB
     400: GOSUB 520: GOTO 220
222  IF A$ = "S" THEN S = 1: GOSUB
     700: GOTO 220
225  IF A$ = "X" THEN 190
226  IF A$ <  > "E" THEN 220
230  GOSUB 260
235  IF S = 1 THEN  VTAB V: HTAB 26
     : PRINT OP(K);"  OUT OF  ";CC(
     K): GOTO 240
236 Y = 2 * V:X = 22: COLOR= 1: IF
     CP(K) <  > 0 THEN  FOR J = 1 TO
     CP(K): PLOT X,Y:X = X + 2: NEXT
     J
237  IF CC(K) <  > 0 AND CC(K) <  >
     CP(K) THEN  COLOR= 15: FOR J =
     1 TO (CC(K) - CP(K)): PLOT X,Y
     :X = X + 2: NEXT J
240  IF CP(K) = T THEN 330
245  NEXT K
250  GOTO 335
251  REM
252  REM   ** EVALUATION SUBROUTINE
      **
256  REM
260 RT$ = R$:G$ =  LEFT$ (G$,T)
265  FOR I = 1 TO T
```

```
270  IF  MID$ (G$,I,1) =  MID$ (R$,
     I,1) THEN CP(K) = CP(K) + 1
275  NEXT
280 J = 0
285  FOR I = 1 TO T
290 J = J + 1
295  IF  MID$ (G$,I,1) <  >  MID$ (
     R$,I,1) THEN  GOTO 300
296 CC(K) = CC(K) + 1: IF J = 1 THEN
     R$ = "" +  RIGHT$ (R$,T - 1): GOTO
     305
297  IF J = T THEN R$ =  LEFT$ (R$,
     T - 1) + "": GOTO 305
298 R$ =  LEFT$ (R$,J - 1) + "" +  RIGHT$
     (R$,T - J): GOTO 305
300  IF J < T THEN 290
305 J = 0
310  NEXT I
315 R$ = RT$: RETURN
316  REM
317  REM   ** CONCLUDING RESPONSES
     **
318  REM
330  GOSUB 380: HTAB 1: VTAB 23: PRINT
     "THAT'S IT!!  CONGRATULATIONS.
      ": FOR I = 1 TO 3: CALL 64484
     : NEXT : GOTO 340
335  GOSUB 380: FOR I = 1 TO 50:A =
      PEEK ( - 16336): NEXT : HTAB
     1: VTAB 23: PRINT "SORRY...   T
     HE ANSWER WAS : ";R$
336  GOSUB 360
```

```
339  HTAB 1: VTAB 24
340  PRINT "PRESS ANY KEY TO CONTIN
     UE. ";: GET A$
345  GOTO 100
360  IF S = 1 THEN  HTAB 2: VTAB 19
     : FOR I = 1 TO T:A$ =  MID$ (R
     $,I,1): PRINT A$;"";: NEXT I: RETURN
365 X = 2:Y = 10 + 2 * G: FOR I = 1
      TO T:A$ =  MID$ (R$,I,1): GOSUB
     500:X = X + 2: NEXT I: RETURN

380  HTAB 1: VTAB 23: PRINT "



              ";: RETURN
400 S = 2: GR
410  COLOR= 1: PLOT 0,39: COLOR= 2:
      PLOT 5,39
415  COLOR= 4: PLOT 10,39: COLOR= 1
     3: PLOT 15,39
420  COLOR= 9: PLOT 20,39: COLOR= 1
     5: PLOT 25,39
425  COLOR= 11: PLOT 30,39: COLOR=
     7: PLOT 35,39
430  IF C <  > 8 THEN  COLOR= 0: HLIN
     (5 * (C - 1) + 1),39 AT 39
435 X = 2:Y = 8
440  COLOR= 5: VLIN 0,35 AT 19
441  HLIN 0,39 AT 8 + 2 * G
443  FOR I = 1 TO 2 * T STEP 2: PLOT
     1 + I,1: PLOT 1 + I,2: PLOT 21
      + I,1: PLOT 21 + I,2: NEXT I
```

```
445  IF K = 0 THEN  RETURN
450  FOR I = 1 TO K
455  FOR J = 1 TO T
460 A$ =  MID$ (X$(I),J,1)
465  GOSUB 500
470 X = X + 2: NEXT J
475 Y = Y + 2
480 X = 2: NEXT I
483  GOSUB 520
485  RETURN
500  IF A$ = "R" THEN  COLOR= 1: PLOT
     X,Y: RETURN
501  IF A$ = "B" THEN  COLOR= 2: PLOT
     X,Y: RETURN
502  IF A$ = "G" THEN  COLOR= 4: PLOT
     X,Y: RETURN
503  IF A$ = "Y" THEN  COLOR= 13: PLOT
     X,Y: RETURN
504  IF A$ = "O" THEN  COLOR= 9: PLOT
     X,Y: RETURN
505  IF A$ = "W" THEN  COLOR= 15: PLOT
     X,Y: RETURN
506  IF A$ = "P" THEN  COLOR= 11: PLOT
     X,Y: RETURN
507  IF A$ = "A" THEN  COLOR= 7: PLOT
     X,Y: RETURN
509  COLOR= 0: PLOT X,Y
510  RETURN
520  IF K = 1 THEN  RETURN
521 Y = 6
522  FOR II = 1 TO K - 1
523 X = 22: COLOR= 1:Y = Y + 2
```

```
525  IF CP(II) <  > 0 THEN  FOR J =
     1 TO CP(II): PLOT X,Y:X = X +
     2: NEXT J
530  IF CC(II) = 0 OR CC(II) = CP(I
     I) THEN  GOTO 538
534  COLOR= 15
535  FOR J = 1 TO (CC(II) - CP(II))
     : PLOT X,Y:X = X + 2: NEXT J
538  NEXT II
540  RETURN
600  HOME : TEXT : VTAB 21: HTAB 1:
      FOR I = 1 TO 8: PRINT A$(I);"
         ";: NEXT
610  PRINT "RED  BLUE  GRN  YLW  OR
     NG  WHT  PINK  AQUA"
620  PRINT : FOR I = 1 TO 16: HTAB
     20: VTAB I: PRINT "****";: NEXT

630  HTAB 1: VTAB 17: PRINT "******
     ******************************
     ****"
640  HTAB 24: VTAB 1: PRINT "CORREC
     T";: HTAB 24: VTAB 2: PRINT "P
     LACEMENT  COLORS";
650  FOR I = 1 TO G: HTAB 20: VTAB
     I + 3: PRINT " "I"  ";: NEXT
660  HTAB 1: VTAB 1: FOR I = 1 TO T
     : PRINT " *";: NEXT I
670  IF C <  > 8 THEN  VTAB 21: HTAB
     (5 * C + 1): FOR I = 1 TO 8 -
     C: PRINT "     ";: NEXT
680  IF C <  > 8 THEN  HTAB (5 * C +
```

```
      1): VTAB 22: FOR I = 1 TO 8 -
      C: PRINT "      ";: NEXT
690  RETURN
700 S = 1
710  TEXT : HOME
720 H = 0:V = 4
725  FOR I = 1 TO K
730  FOR J = 1 TO T
735 H = H + 2: HTAB H: VTAB V: PRINT
       MID$ (X$(I),J,1);
740  NEXT J
750 H = 0:V = V + 1: NEXT I
755 V = 4
760  IF K <  > 1 THEN  FOR I = 1 TO
      K - 1: VTAB V: HTAB 26: PRINT
      CP(I);"    OUT OF  ";CC(I);:V =
      V + 1: NEXT
770  RETURN
```

程式說明

表 2-3 副程式列表

副 程 式	行號
時間延遲迴路	10
文字幕格式	600
圖形顯示格式	400
項目列表	90
游標顯示	25
答案鑑別	260
最終結果顯示	325
由彩色圖形至文字幕之間變換	700

程式由第五行開始，爲了做顏色與文字代號間變換，因此做了維數（DIM）的宣告。雖然APPLE的游標在文字幕時是一個醒目的閃爍方塊，可是在彩色圖形時並不能產生這種游標，因此由 16 行至 56 行來做出一個閃爍的彩色方塊。

主程式是由第100行開始的，先做出標題畫面，延時了數秒鐘之後開始由第105行、第115行、第125行來要求操作者選擇顏色的數目、待塡位置的數目、以及最大允許的做答次數，第131行選擇是否彩色圖案還是用文字幕顯示，此地POKE 50 ，127是造出一閃爍字形，而POKE 50 ，255是還原成正常字幕，第132行至166行用以造出任意的顏色排列以便參加遊戲者猜題，同時依變數S來判斷是否跳至400行或600行的格式副程式。第180至215行將遊戲者的答案保存，之後程式轉移至項目選擇的副程式，此副程式要求遊戲者選擇E（送入電腦）、X（改變答案從新答起）、S（改變顯示方式）。如果鍵入E鍵，則程式轉至第255行至315行判別答案並顯示出來。如果做答已至原先預定次數而仍未答對，則由320至345行將正確答案印出。其中330行的CALL 64484是呼叫喇叭發聲副程式。

程式改進

當選擇使用彩色方塊來做畫面顯示的時候，程式執行後發現做答時按鍵的動作不太理想，有時必須按兩三次才能夠將答案輸入，這是由於游標圖形顯示迴路 25 行至 56 行速度太慢的緣故，可將 16 行至第 25 行消去，並且加入：

```
199  IF S＝2 THEN HTAB H： VTAB V： GET
     A$： COLOR＝0： X＝H： Y＝2＊V： GOSUB
     500： GOTO 210
```

這樣的話，當選擇彩色圖形顯示時，游標是不閃爍的，而且出現的地方是在下一個將要塡入顏色位置的左上方，這時試試看按鍵反應情形，前述的

缺點已不存在了。

執行例

當程式 RUN 之後，即出現圖2-9之畫面，接下來詢問遊戲者要猜多少顏色，要填入幾個位置，最多允許猜多少次？最後並要求選擇以彩色畫面來顯示或是以文字幕顯示，如圖 2-10 所示。

```
]

]RUN
**   COLOR   **
**   CODE    **
**   COMBO   **
```

圖 2-9

```
HOW MANY COLORS DO YOU
WANT TO WORKWITH (3-8)   4

HOW LARGE A TABLEAU (3-8)   2
?REENTER
HOW LARGE A TABLEAU (3-8)   4

HOW MANY GUESS WILL YOU NEED   (4-12)6

WHITCH MODE, COLOR ORALPHABET?
```

圖 2-10

程式六 跳 蛙

這是一個簡單的遊戲，可以練習判斷思考方式。

使用説明

如圖2-11(a)所示，遊戲的目的是要將圖中的“○”和“×”的位置對調，對調的方式是所欲移動的位置，若鄰位置是空格則移一格，若次鄰位置是空格，則跳一格。如圖2-11(b)所示（注意“○”不能跳過“○”，“×”不能跳過“×”）。

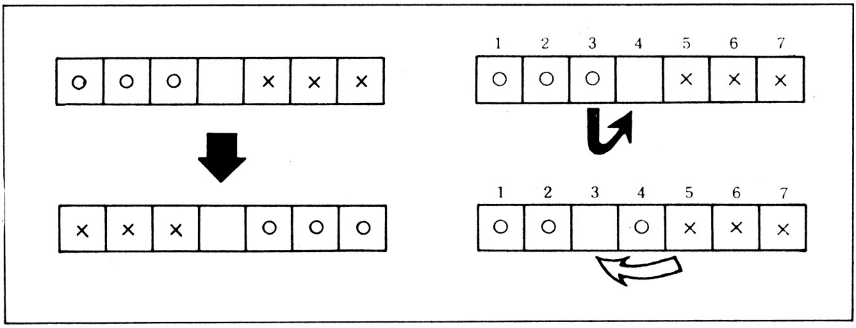


圖 2-11(a)　　圖 2-11(b)

程 式

```
] LIST
10   REM****************
11   REM** LEAP FROG **
12   REM****************
13   PRINT
14   PRINT "*** LEAP FROG ***"
20   DIM X$(7)
25   DIM B(4)
40   GOSUB 500
45   PRINT TAB(5);"(1234567)"
50   GOSUB 100
55   GOSUB 400
60   GOSUB 150
70   GOSUB 400
75   N=N+1
80   IF M=1 THEN 600
90   GOTO 60
100  FOR I=1 TO 3
110  X$(I)="O": X$(I+4)="X"
120  NEXT I
130  X$(4)=" "
135  B(1)=1:B(2)=-1:B(3)=2:B(4)
     =-2
140  RETURN
150  PRINT TAB(25);"WHICH ONE? "
```

```
160 INPUT I
165 IF I=0 THEN 630
170 FOR J2=1 TO 4 FOR J2=1 TO 4
172 J=I+B(J2)
173 IF J<1 OR J>7 THEN 180
174 IF X$(J)=" " THEN 190
175 IF J<1 OR J>7 THEN 180
176 IF X$(J)=" " THEN 190
180 NEXT J2
185 GOTO 240
190 IF ABS(I-J)=1 THEN 220
200 J1=(I+J)/2
210 IF X$(J1)=X$(I) THEN 240
220 X$(J)=X$(I) : X$(I)=" "
230 GOTO 270
240 PRINT "        CAN'T BE MOVED"
250 GOTO 150
270 FOR K=1 TO. 3
280 IF X$(K)="X" AND X$(K+4)= "O"
    THEN 300
290 GOTO 320
300 NEXT K
310 IF X$(4)=" " THEN M=.1
320 RETURN
400 REM ＊＊＊＊ PRINT ＊＊＊＊
420 PRINT TAB(5); "(";
430 FOR L=1 TO 7
440 PRINT " "; X$(L);
```

```
450 NEXT L
460 PRINT " ) " ;
470 RETURN
500 PRINT
510 PRINT "( O O O   X X X ) ==>
    ( X X X   O O O )"
520 PRINT :INVERSE
530 PRINT "  RULE  " : NORMAL
540 PRINT "  THIS IS A NICE SIMP LE
    PROGRAM, TESTING THE PLAY ER'S
    LOGIC."
550 PRINT
560 PRINT "THE OBJECT IS TO MOVE ALL
    THE FROGS ON THE LEFT TO RIGHT,
    AND VICE VERSA."
570 PRINT
580 PRINT "FROGS CAN ONLY LEAP OVER
    ONE FROGOR MOVE TO AN AD JACENT
    EMPTY SPACE."
585 PRINT
590 RETURN
600 PRINT : PRINT : PRINT "      .... YOU
    GET ....      " : PRINT
610 PRINT "YOU DID IT IN";N;"STEPS"
620 GOTO 999
999 END
```

執行例

```
(1 2 3 4 5 6 7)          ]RUN
(O O O   X X X)  WHICH ONE? 3
(O O   O X X X)  WHICH ONE? 5
(O O X O   X X)  WHICH ONE? 4
(O O X   O X X)  WHICH ONE? 6
(O O X X O   X)  WHICH ONE? 5
(O O X X   O X)  WHICH ONE? 4
(O O X   X O O)  WHICH ONE? 2
(O   X O X O X)  WHICH ONE? 3
(O X   O X O X)  WHICH ONE? 1
(  X O O X O X)  WHICH ONE? 2
(X   O O X O X)  WHICH ONE? 4
CAN'T BE MOVED
(X O   O X O X)  WHICH ONE? 5
(X O X O   O X)  WHICH ONE? 7
(X O X O X O  )  WHICH ONE? 6
(X O X O X   O)  WHICH ONE? 4
(X O X   X O O)  WHICH ONE? 2
(X   X O X O O)  WHICH ONE? 3
(X X   O X O O)  WHICH ONE? 5
(X X X O   O O)  WHICH ONE? 4
(X X X   O O O)
....YOU GET....
YOU DID IT IN 19 STEPS
```

圖 2-12

程式七 分解因數

要將任意整數分解成爲由質數構成的連乘積，以人力來完成的話却也不是一件容易的事，但是以簡短的程式用電腦來處理，却是一件輕而易舉的事情，請將下列程式鍵入，然後試試看其效力大不大？

程 式

```
]LIST

0  TEXT : HOME
10  PRINT : INVERSE
20  PRINT "*** START OF THE PROGRAM
     ***": NORMAL
30  INPUT "INPUT AN INTEGER NUMBER"
    ;A
40  PRINT
50  LET A1 = A
60  IF A = 0 THEN 999
70  IF A = 2 THEN 390
90  LET Q = 0
100  IF A > 0 THEN 130
110  FLASH : PRINT "THIS NUMBER < 0
     !!!---STOP": NORMAL
120  GOTO 999
```

```
130  LET P = 2
140  GOSUB 180
150  FOR P = 3 TO  SQR (A) STEP 2
160  GOSUB 180
170  GOTO 340
180  LET R = 0
190  IF A / P =  INT (A / P) THEN 2
     10
200  GOTO 240
210  LET A = A / P
220  LET R = R + 1
230  GOTO 190
240  IF R < 1 THEN 330
250  IF Q = 1 THEN 320
260  LET Q = 1
265  PRINT "***********************
     "
270  PRINT A1;" IS RESOLVED TO BE:"

280  PRINT "***********************
     "
290  PRINT "FACTOR","POWER"
300  PRINT "------","-----"
320  PRINT P,R
330  RETURN
340  NEXT P
350  IF A = 1 THEN 10
360  IF Q = 0 THEN 390
370  PRINT A,1
375  PRINT
380  GOTO 10
390  PRINT A;" IS A PRIME"
```

```
400  GOTO 10
999  END
```

流程圖

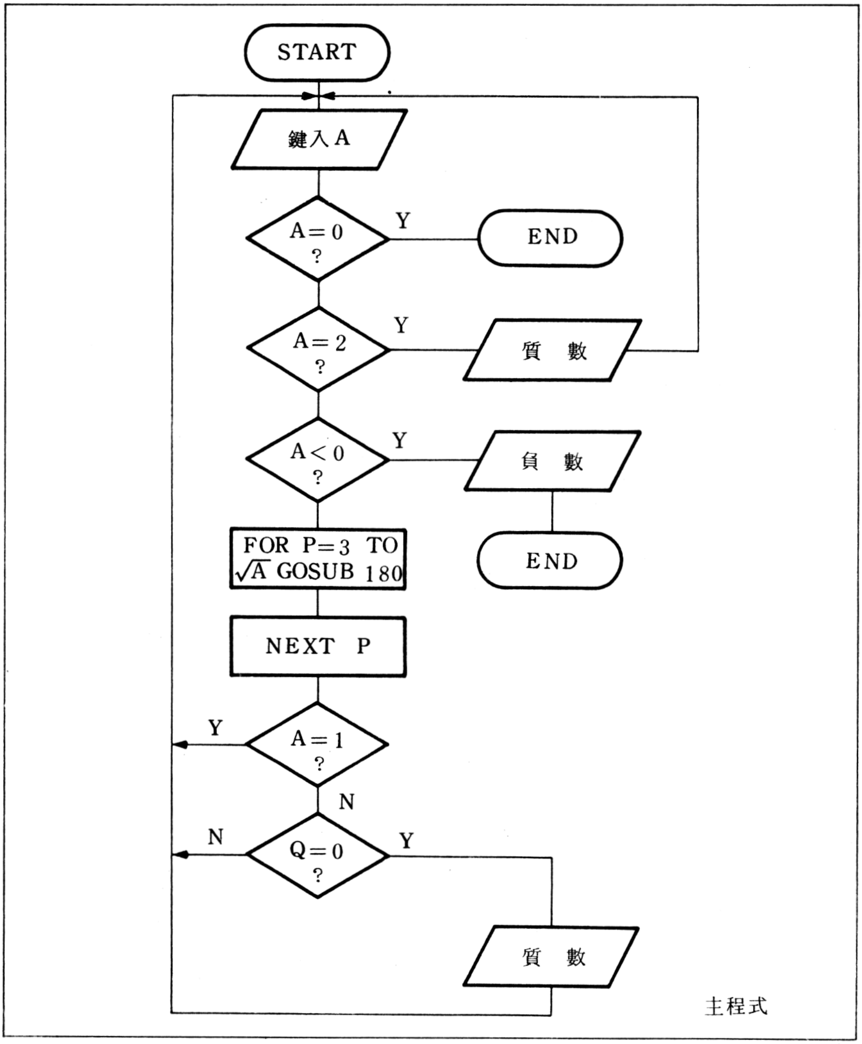
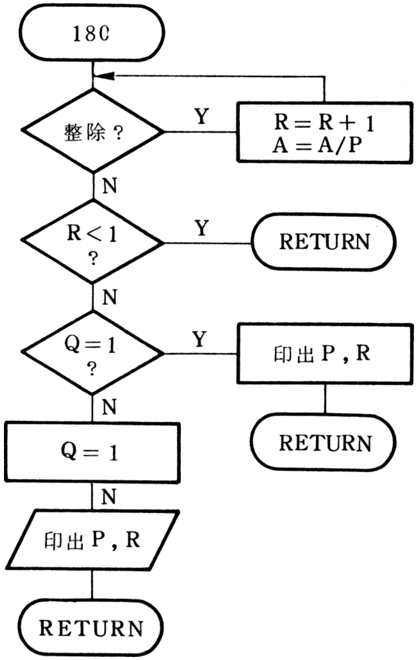


圖 2-13(a)



180副程式

圖 2-13(b)

執行例

程式執行後，螢幕印出“ START OF THE PROGRAM ”的字樣，並要求輸入一個整數（ INPUT AN INTEGER NUMBER ），在本例中，首先送入 12 這個數，結果就分解成 $2^2 \times 3^1 = 12$ ，螢幕上的排列 FACTOR 是指該數的因子，POWER是指該因子的次方，接下來輸入 4620 ，結果是 $2^2 \times 3^1 \times 5^1 \times 7^1 \times 11^1 = 4620$ 。

如果鍵入的數是一個負數，則螢幕印出警告字樣，而且程式停止執行，此時必須重新 RUN 才行。如果鍵入的是一質數，則該數不分解而直該告訴使用者“ 這是一個質數 ” 。

```
]RUN
*** START OF THE PROGRAM ***
INPUT AN INTEGER NUMBER12

************************
12 IS RESOLVED TO BE:
************************
FACTOR              POWER

------              -----
2                   2
3                   1
*** START OF THE PROGRAM ***
INPUT AN INTEGER NUMBER4620

************************
4620 IS RESOLVED TO BE:
************************
FACTOR              POWER

------              -----
2                   2
3                   1
5                   1
7                   1
11                  1
```

圖 2-14(a)

```
*** START OF THE PROGRAM ***
INPUT AN INTEGER NUMBER3072

************************
3072 IS RESOLVED TO BE:
************************
FACTOR              POWER

------              -----
2                   10
3                   1
*** START OF THE PROGRAM ***
INPUT AN INTEGER NUMBER-1

THIS NUMBER < 0!!!---STOP

]RUN
*** START OF THE PROGRAM ***
INPUT AN INTEGER NUMBER173

173 IS A PRIME
*** START OF THE PROGRAM ***
INPUT AN INTEGER NUMBER11

11 IS A PRIME
*** START OF THE PROGRAM ***
```

圖 2-14(b)

程式八　學習英語字母

這是一個簡單的程式，可以用來教小孩子學習 26 個英文字母。使用者可在程式中加入其他如：阿拉伯數字、英文單字等的練習。

首先，程式開始執行，字幕上會將 26 個英文字母一個個的依次顯示出來，每兩個字母顯示之間空了一段時間間隔，以便學習者用來認識該字母，如果教學者最初幾次能在旁輔助教以發音，則更能增加學習效果。當 26 個字母全部顯示完畢之後，畫面停止，使學習者能依需要在字幕上學習每一個字母，直到自認爲熟悉爲止。此時只要按下“ RETURN ”鍵，則畫面上開始以不定的次序將一個字母遮蓋住，並且發出閃爍訊號，要求學習者由鍵盤上找出該字母並鍵入。如果答對了，則繼續遮蓋住另外一個字母由學習者作答，若答錯了，則由喇叭發出“嗶”的響聲，並要求再另作答。

程　式

```
]LIST

20  HOME
21  INVERSE : HTAB 12: PRINT "TEACH
      YOUR CHILDERN": NORMAL
30  PRINT
40  PRINT "THIS PROGRAM IS DESIGNED
      TO HELP YOUR YOUNG-ONE"
43  PRINT "RECOGNIZE THE 26 DIFFERE
```

```
    NT LETTERS OF THE ALPHABET."
45  PRINT "MOREOVER, IT WILL TEACH
    THEM TO RECOGNIZE 10"
50  PRINT
55  INPUT "PLEASE TYPE YOUR CHILDS
    NAME.";NA$
100  HOME
101  PRINT : PRINT "HELP YOUR CHILD
      PRONOUNCE THE LETTERS AS THEY
     'RE PRINTED."
102 R = 6000: GOSUB 300
110  DIM B$(26)
111  GOSUB 330
135  FOR V = 0 TO 3: PRINT " ";: HTAB
     40: PRINT " ";: NEXT V
190 R = 3000: REM  DISPLAY  SPEED
200  FOR I = 1 TO 26
210  READ B$(I): NEXT
220 VV = 5: GOSUB 320: GOSUB 330
240  PRINT : PRINT
250  PRINT "STUDY THE ALPHABET, WHE
     N READY PRESS 'RETURN' KEY"
255 Z =  PEEK ( - 16384): IF Z <  >
     141 THEN 255
256  POKE  - 16368,0
257  HOME : GOSUB 310
258 I =  INT ( RND (1) * 26) + 1
260  GOSUB 340
269  INVERSE
270  VTAB 8: HTAB 4: INPUT "WHAT LE
     TTER?";L$: IF L$ = B$(I) THEN
     280
```

```
274  FLASH
275  VTAB 10: HTAB 4: PRINT "THINK
     AGAIN!   ";"": REM  CONTROL 'G

276  GOTO 269
280  VTAB 10: HTAB 4: PRINT "YOU AR
     E RIGHT!!"
290  INVERSE :R = 0: GOSUB 310: GOTO
     258
300  FOR Z = 0 TO R: NEXT : RETURN

310  RESTORE : FOR I = 1 TO 26: READ
     B$(I): NEXT
320  FOR I = 1 TO 17: HTAB 2 + I *
     2: VTAB (VV): PRINT B$(I): GOSUB
     300: NEXT
321  FOR I = 18 TO 26: HTAB I * 2 -
     24: VTAB VV + 2: PRINT B$(I): GOSUB
     300: NEXT : RETURN
330  HTAB 1: INVERSE : FOR X = 0 TO
     39: PRINT " ";: NEXT : RETURN
340  FLASH : IF I > 17 THEN  HTAB I
      * 2 - 24: VTAB VV + 2: PRINT
     " ": RETURN
341  HTAB 2 + I * 2: VTAB VV: PRINT
     " ": RETURN
350  DATA A,B,C,D,E,F,G,H,I,J,K,L,M
     ,N,O,P,Q,R,S,T,U,V,W,X,Y,Z
375  PRINT
```

註：275行中的REM CONTROL 'G' 是指“ ”內包含了一個CTRL G

程式改進

如果將DATA的內容增加（譬如阿拉伯數字或英文單字），就可以增加學習的內容。程式中當每一個字母出現時，加入字母發音的副程式，如此可以使教學更自動化，也更生動了。

程式九　行星軌道

使用過APPLE電腦的人一定會發現，和其它形式的微電腦比較起來，在畫面上的顯示方式是有顯着的不同。在某些微電腦的鍵盤上，備有特殊的圖形或符號鍵，甚或小寫英文與日文字鍵，因此可以方便地將該圖形或字母鍵入程式裡頭來使用。而APPLE是沒有這種設計的，其文字與圖形也不能混合在一起。雖然看起來，這似乎是一個缺點，可是APPLE的高解像度畫面却可彌補這種缺點，尤其是經過適當設計的程式，更可發揮其他微電腦所缺乏細緻圖像的優點。

APPLE的高解像畫面的長處，通常可在畫函數圖形、曲線等表現出來，此處所舉的例子是以電腦來模擬太陽系的行星運動情形，程式執行時可以給我們一個對於太陽系組織的概念。

程　式

```
]LIST

5  TEXT : HOME : CLEAR
10  DIM OP(9),PY(9),I(9)
50  GOTO 300
80  HOME : VTAB 21
90  INVERSE : PRINT "    **  A P P L
     E     O R R E R Y  **    ": NORMAL
```

```
100  REM  **  PLOT CIRCULAR ORBITS
105  REM  **  OFFSET PLUTO'S ORBIT
     TO SIMULATE ELLIPSE
110  INVERSE : PRINT "DAY
         YEAR  (EARTH)         ": NORMAL

115  HGR : HCOLOR= 3: POKE 34,22
120  LET W = PI / 180
130  FOR J = 1 TO 9
131  LET R = OP(J) / SCALE
135  LET I(J) = I(J) + (PY(3) / PY(
     3) * TD)
140  LET Y = Z * I(J)
150  LET X = XC + R *  COS (W + Y)
160  LET Y = YC - R *  SIN (W + Y)
165  IF J = 9 THEN  LET X = X + (15
     00E06 / SCALE)
170  IF X > 279 OR X < 0 THEN  GOTO
     190
175  IF Y > 159 OR Y < 0 THEN  GOTO
     190
180  HPLOT X,Y
190  NEXT J: IF RBIT = 1 THEN  GOTO
     200
195  CALL 62450
200  HPLOT XC - 1,YC - 1 TO XC + 1,
     YC
210  HPLOT XC,YC - 1 TO XC,YC + 1
215  PRINT I(3),I(3) / PY(3)
220  GOTO 130
300  REM  **  SET UP SOLAR SYSTEM
500  REM  **  OP(I)=DIAMETER OF PLA
```

```
     NETARY ORBIT
505  REM  **  PY(I)=PLANETARY YEAR
510  LET OP(1) = 5.79E06
511  LET PY(1) = 87.97
520  LET OP(2) = 108.2E06
521  LET PY(2) = 224.7
530  LET OP(3) = 149.6E06
531  LET PY(3) = 365.26
540  LET OP(4) = 227.9E06
541  LET PY(4) = 686.98
550  LET OP(5) = 778.3E06
551  LET PY(5) = 11.86 * PY(3)
560  LET OP(6) = 1427E06
561  LET PY(6) = 29.46 * PY(3)
570  LET OP(7) = 2869.6E06
571  LET PY(7) = 84.01 * PY(3)
580  LET OP(8) = 4496.6E06
581  LET PY(8) = 164.79 * PY(3)
590  LET OP(9) = 5.9E09
591  LET PY(9) = 248.4 * PY(3)
600  LET PI = 3.14159
601  LET P2 = PI * 2
650  FOR J = 0 TO 9:I(J) = 0: NEXT
     J
700  LET Z = P2 / 360
800  INPUT "SCALE=(5E5--1E8)";SCALE

820  PRINT "PLOT ORBITAL PATH (TYPE
      '1') OR"
825  INPUT "PLANET POSITION ONLY(TY
     PE '2') ";RBIT
850  INPUT "SUN LOCATION (X.1--278;
```

```
      Y,1--158)? ";XC,YC
875  INPUT "TIME INCREMENT(DAYS)";T
     D
1000  GOTO 80: END
```

程式説明

本程式使用APPLESOFT BASIC語言以及高解像畫面（HIRES）來組成一個簡單的行星軌道模型。把這個初步模型加以改進後，可以造成更實用、更精確的效果。無論是個人練習或是教學、應用上皆可派上用場。

爲了簡化問題，首先假設行星軌道是圓形的，因爲已經將星際距離縮小在電視螢光幕上，所以原本接近於圓的橢圓軌道以圓形來加以近似，其差異是看不出來的。其次，任何行星，無論其大小均用一個點來代表，由於也是尺寸縮小的緣故，地球、土星、木星大小的差異也就看不出來了。另外由於各行星的月亮，因爲與各該行星的距離太接近，因此就與行星合爲一體而分辨不出了。

執行本程式時，可由使用者來選擇尺度的大小。這是因爲星際距離太大往往了爲觀測近太陽的行星而不能觀測到遠日行星（已經在螢幕之外了），而觀測遠日行星，又不能分辨近日行星（因比例縮得太小而與太陽合在一起了），所以程式執行時會提醒使用者選擇適當的尺度。而因此更可令使用者能對太陽系的尺度大小有一清楚的概念。通常欲觀測冥王星，則近太陽軌道上的行星是無法觀測的，而觀測近太陽行星軌道時，木星以外的行星是看不到的。

選取尺度大小之後，程式會再提醒使用者選取太陽的位置，太陽的位置可以在螢幕上158×278的範圍內，而不是一定在中央點上。譬如將太陽選擇在螢幕右上角，則可以觀測到四分之一的軌道象限，但軌道半徑却加倍了。

程式開始執行後，各行星依其軌道與各別的速度繞太陽運轉，使用者

可以選擇將運行軌跡完全掃描出來，亦或是只將每一時刻的位置點出兩種畫面方式。還可以選擇觀測的時間間隔（即每點間的時間間隔），通常內圍軌道需要較缺的時間間隔才能描繪完整，而外圍則要較長的時間間隔。

時間的單位是以地球上的日或年為單位。而距離的尺度則由 5E5 = 5×10^{5} 至 1E8 = 1×10^{8} 之間來做選擇，譬如選 1×10^{6} = 1E6 為單位，則地球至太陽的位置 149.6×10^{6} km在螢幕上就是 149.6≈150 個間隔。

程式改進

因為這是相當簡單的模擬，因此有待改進的地方很多。譬如，可以將圓形軌道改成橢圓軌道，在第150行及第160行的R當使用橢圓軌道的時候，就要用 R1 及 R2 來橢圓的長軸及短軸。另外加入刻卜勒定律，要求在相同時間所掃描之面積相等亦可。如果考慮到每個星球的月亮時，可以經由畫面將所欲觀察的行星加以放大，或是指定單獨一個行星來顯示。如此程式可能變為複雜，但却更詳盡了。

執行例

```
]
]RUN
SCALE=(5E5--1E8)1E7
PLOT ORBITAL PATH (TYPE '1') OR
PLANET POSITION ONLY(TYPE '2') 1
SUN LOCATION (X,1--278; Y,1--158)? 100,100
TIME INCREMENT(DAYS)1
   \\  A P P L E    O R R E R Y  \\
DAY                 YEAR  (EARTHD
1                 2.7377583E-03
2                 5.4755516SE-03
3                 8.21332749E-03
4                 .0109511033
```

5	.0136888792
6	.016426655
7	.0191644308
8	.0219022066
9	.0246399825
10	.0273777583
11	.0301155341
12	.03285331

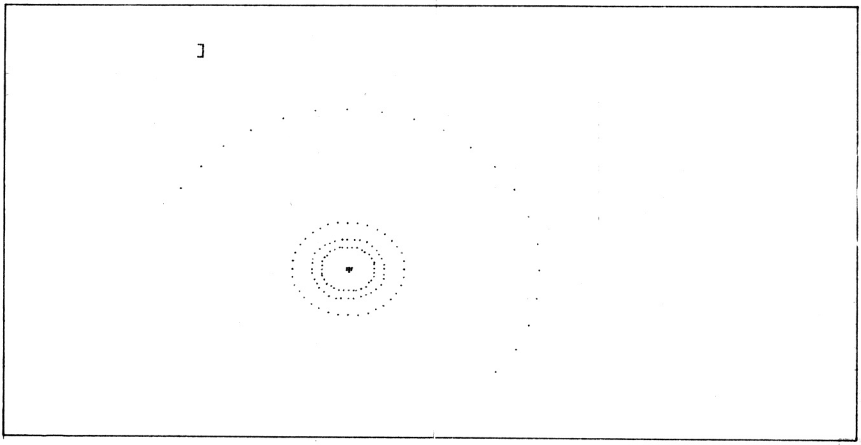


圖 2-15

程式十　代數(解多元一次方程式)

本程式是用來將多元一次方程式化簡，並將化簡的過程列出來，所以當做教學工具來使用是很恰當的。使用者可將任意多元一次的方程式送入電腦，其運算過程及結果將會顯示在螢幕上。將此程式當成電腦輔助教學的例子（C.A.E 即 COMPUTER AID EDUCATION 電腦輔助教育之意），吾人可由此探究 C.A.E 的簡單應用而加以發揮。

程式本身相當明瞭易懂，只不過一直執行一連串重複的工作，因此有相當多類似處（如第 1 行至第 55 行即是），所以此處只列出程式而不加以一項一項的說明，程式中並不用到什麼特殊技巧，相信讀者會很容易自行解讀的。

程　式

```
]LIST

0  GOSUB 22000: GOTO 40020
1 AA = AA + UN: RETURN
2 BB = BB + UN: RETURN
3 CC = CC + UN: RETURN
4 DD = DD + UN: RETURN
5 EE = EE + UN: RETURN
6 FF = FF + UN: RETURN
7 GG = GG + UN: RETURN
```

```
8 HH = HH + UN: RETURN
9 II = II + UN: RETURN
10 JJ = JJ + UN: RETURN
11 KK = KK + UN: RETURN
12 LL = LL + UN: RETURN
13 MM = MM + UN: RETURN
14 NN = NN + UN: RETURN
15 OO = OO + UN: RETURN
16 PP = PP + UN: RETURN
17 QQ = QQ + UN: RETURN
18 RR = RR + UN: RETURN
19 SS = SS + UN: RETURN
20 TT = TT + UN: RETURN
21 UU = UU + UN: RETURN
22 VV = VV + UN: RETURN
23 WW = WW + UN: RETURN
24 XX = XX + UN: RETURN
25 YY = YY + UN: RETURN
26 ZZ = ZZ + UN: RETURN
30 DV = AA:AA = 0: RETURN
31 DV = BB:BB = 0: RETURN
32 DV = CC:CC = 0: RETURN
33 DV = DD:DD = 0: RETURN
34 DV = EE:EE = 0: RETURN
35 DV = FF:FF = 0: RETURN
36 DV = GG:GG = 0: RETURN
37 DV = HH:HH = 0: RETURN
38 DV = II:II = 0: RETURN
39 DV = JJ:JJ = 0: RETURN
40 DV = KK:KK = 0: RETURN
41 DV = LL:LL = 0: RETURN
42 DV = MM:MM = 0: RETURN
```

```
43 DV = NN:NN = O: RETURN
44 DV = OO:OO = O: RETURN
45 DV = PP:PP = O: RETURN
46 DV = QQ:QQ = O: RETURN
47 DV = RR:RR = O: RETURN
48 DV = SS:SS = O: RETURN
49 DV = TT:TT = O: RETURN
50 DV = UU:UU = O: RETURN
51 DV = VV:VV = O: RETURN
52 DV = WW:WW = O: RETURN
53 DV = XX:XX = O: RETURN
54 DV = YY:YY = O: RETURN
55 DV = ZZ:ZZ = O: RETURN
85  FOR Z1 = 1 TO 2000: NEXT Z1: HOME
     : PRINT "INSTRUCTIONS ";: GET
     E$: IF E$ <  > "Y" THEN 21150
86  GOSUB 40030
90  DIM A$(100),ALPHABET$(26):ALPHA
     BET$ = "ABCDEFGHIJKLMNOPQRSTUV
     WXYZ": PRINT "SOLVING FOR ";: GET
     Z$: PRINT Z$: PRINT : INPUT A$
     :L =  LEN (A$)
95  IF A$ = "" THEN  HOME : VTAB (3
     ): INVERSE : PRINT "BYE FOR NO
     W": NORMAL : END
150  FOR O = 1 TO L
160  IF  MID$ (A$,O,1) = "=" THEN E
     QUAL = O
180  NEXT O
190  IF EQUAL = 0 THEN  GOTO 700
210  FOR P = EQUAL TO L
220  IF  MID$ (A$,P,1) = "(" THEN  GOSUB
```

```
     20000
230  IF  MID$ (A$,P,1) = ")" THEN  GOSUB
     20020
240  NEXT P
250  IF RA <  = 0 THEN PJ = L:F2 =
     PJ:P = EQUAL:F3 = P:MULT = 1:M
     I = 1: GOSUB 20180: PRINT  LEFT$
     (A$,EQUAL);: GOSUB 20550: GOSUB
     21090: GOSUB 21210: PRINT : GOTO
     360
260  IF RA <  > RE THEN  PRINT "PAR
     ENTASIS DON'T MACH ON RIGHT SI
     DE OF": PRINT "EQUATION": END

270 PJ = RPA(1) - 1:P = EQUAL: GOSUB
     20080:SW = Q - 1: IF  MID$ (A$
     ,Q,1) = "=" THEN SW = SW + 1
280  GOSUB 20360: IF T(2) = 1 AND M
     (2) = 1 THEN MULT = T(1):MI =
     M(1): GOSUB 20160:F2 = PJ:F3 =
     P: GOSUB 20180: PRINT  LEFT$ (
     A$,SW);: GOSUB 20550: GOSUB 21
     090: PRINT  MID$ (A$,RUR(1) +
     1,L): GOTO 320
290  IF T(3) = 1 AND M(3) = 1 THEN
      GOTO 312
295 MULT = T(3):MI = M(3): GOSUB 20
     160:F2 = PJ:F3 = P: GOSUB 2018
     0: PRINT  LEFT$ (A$,RPA(2));: GOSUB
     20550: GOSUB 21090
296  PRINT  MID$ (A$,RUR(1) + 1,L):
      GOSUB 20490: IF P <  > PJ THEN
```

```
    MULT = 1:MI = 1:F2 = PJ:F3 = P
    : GOSUB 20180: PRINT  LEFT$ (A
    $,RPA(2));: GOSUB 20550: GOSUB
    21090: PRINT  MID$ (A$,RUR(2),
    L)
297 GOSUB 21115:MULT = T(3) * T(2)
    :MI = M(3) * M(2): GOSUB 20160
    :F2 = PJ:F3 = P: GOSUB 20180: GOSUB
    20490
298 IF P <  > PJ THEN F2 = PJ:F3 =
    P:MULT = T(2):MI = M(2): GOSUB
    20180
299 PRINT  LEFT$ (A$,RPA(1));: GOSUB
    20550: GOSUB 21090: PRINT  MID$
    (A$,RUR(2) + 1,L): GOSUB 20520
    : IF P <  > PJ THEN F2 = PJ:F3
     = P:MULT = 1:MI = 1: GOSUB 20
    180: PRINT  LEFT$ (A$,RPA(1));
    : GOSUB 20550: GOSUB 21090: PRINT
     MID$ (A$,RUR(3),L)
300 GOSUB 21115:MULT = T(1) * T(2)
     * T(3):MI = M(1) * M(2) * M(3
    ): GOSUB 20160:F2 = PJ:F3 = P:
     GOSUB 20180:MULT = T(1) * T(2
    ):MI = M(1) * M(2): GOSUB 2049
    0: IF P <  > PJ THEN F2 = PJ:F
    3 = P: GOSUB 20180:MULT = T(1)
    :MI = M(1): GOSUB 20520: IF P <
     > PJ THEN F2 = PJ:F3 = P: GOSUB
    20180
310 PRINT  LEFT$ (A$,SW);: GOSUB 2
    0550: GOSUB 21090: PRINT  MID$
```

```
     (A$,RUR(RE) + 1,L): GOTO 320
312 MULT = T(2):MI = M(2): GOSUB 20
     160:F2 = PJ:F3 = P: GOSUB 2018
     0: PRINT  LEFT$ (A$,RPA(1));: GOSUB
     20550: GOSUB 21090: PRINT  D$
     (A$,RUR(1) + 1,L)
313  GOSUB 20490: IF P <  > PJ THEN
     MULT = 1:MI = 1: PRINT  LEFT$
     (A$,RPA(1));:F2 = PJ:F3 = P: GOSUB
     20180: GOSUB 20550: GOSUB 2109
     0: PRINT  MID$ (A$,RUR(2),L)
314  GOSUB 21115:MULT = T(1) * T(2)
     :MI = M(1) * M(2): GOSUB 20160
     :F2 = PJ:F3 = P: GOSUB 20180:M
     ULT = T(1):MI = M(1): GOSUB 20
     490:F2 = PJ:F3 = P: IF F2 <  >
     F3 THEN  GOSUB 20180
319  PRINT  LEFT$ (A$,SW);: GOSUB 2
     0550: GOSUB 21090: PRINT  MID$
     (A$,RUR(2) + 1,L)
320 MULT = 1:MI = 1:PJ = RPA(1) - 1
     :P = EQUAL: GOSUB 20080:SW = Q
      - 1: IF  MID$ (A$,Q,1) = "=" THEN
     SW = SW + 1
330 PJ = L:F2 = PJ:F3 = RUR(RE): IF
     F2 <  > F3 THEN  GOSUB 20180: PRINT
      LEFT$ (A$,SW);: GOSUB 20550: GOSUB
     21090: GOSUB 21210: PRINT
335 PJ = SW:F2 = PJ:F3 = EQUAL: IF
     F2 <  > F3 THEN  GOSUB 20180: PRINT
      LEFT$ (A$,EQUAL);: GOSUB 2055
     0: GOSUB 21090: GOSUB 21210: PRINT
```

```
360  GOSUB 21205: IF K3 <  > 1 THEN
      PRINT  LEFT$ (A$,EQUAL - 1);:
      GOSUB 21160: INVERSE : GOSUB
     20550: GOSUB 21160: PRINT "=";
     : GOSUB 20550: GOSUB 21160: INVERSE
     : GOSUB 20550: GOSUB 21160: GOSUB
     21090: GOSUB 21190: GOSUB 2121
     0: PRINT
362 X4 = 1:PJ = EQUAL - 1:F2 = PJ:P
      = 0:F3 = P: GOSUB 21110: GOSUB
     20180: GOSUB 21130
363  GOSUB 20550:JO = JO *  - 1: GOSUB
     21090: PRINT "=";: GOSUB 21750
     :JO = JO *  - 1: GOSUB 21230: PRINT

364  IF JO <  > 0 THEN  GOSUB 20550
     :JO = JO *  - 1: GOSUB 21090: INVERSE
     :JO = JO *  - 1: GOSUB 21090: PRINT
     "=";: GOSUB 21750: INVERSE : GOSUB
     21090: GOSUB 21230: PRINT
365 JO = JO + JA: GOSUB 20550: PRINT
     "=";: GOSUB 21090: IF JO = 0 THEN
      PRINT "0";
370  FOR F = 1 TO 26
375  IF Z$ =  MID$ (ALPHABET$,F,1) THEN
      GOTO 380
376  NEXT F: GOTO 21150
380  PRINT : ON F GOSUB 30,31,32,33
     ,34,35,36,37,38,39,40,41,42,43
     ,44,45,46,47,48,49,50,51,52,53
     ,54,55
385  IF DV = 0 THEN  GOTO 21150
```

```
390  GOSUB 21205: IF K3 <  > 1 THEN
      PRINT DV;: PRINT Z$;: GOSUB 2
     0550: GOSUB 21160: INVERSE : GOSUB
     20550: PRINT "=";: INVERSE : GOSUB
     20550: GOSUB 21090: GOSUB 2121
     0: PRINT : GOTO 392
391  GOSUB 21160
392  PRINT DV;: PRINT Z$;: PRINT "=
     ";: GOSUB 20550: GOSUB 21090: GOSUB
     21170: GOSUB 21190: IF K2 <  >
     1 THEN  PRINT : PRINT Z$;: PRINT
     "=";: GOSUB 20550: GOSUB 21090
     : GOTO 21150
400  PRINT "0": GOTO 21150
700 MULT = 1:MI = 1:P = 0:F3 = P:PJ
      = L:F2 = PJ: GOSUB 20180: GOSUB
     20550: GOSUB 21090: GOSUB 2119
     0: IF K2 = 1 THEN  PRINT "+0";


710  GOTO 21150
20000 RA = RA + 1: IF RA >  = 4 THEN
       GOTO 40000
20010 RFA(RA) = P: RETURN
20020 RE = RE + 1: IF RE >  = 4 THEN
       GOTO 40010
20030 RUR(RE) = P: RETURN
20040  FOR NA = 1 TO 26
20050  IF  MID$ (A$,PJ,1) =  MID$ (
      ALPHABET$,NA,1) THEN  RETURN
20060  NEXT NA
20070 NA = 0: RETURN
20080 D3 = 0:U7 = 0
```

```
20090  FOR Q = PJ TO P STEP  - 1
20095 D3 = D3 + 1
20099  IF Q = 0 THEN  RETURN
20100  IF  MID$ (A$,Q,1) = "+" THEN
       RETURN
20110  IF  MID$ (A$,Q,1) = "-" THEN
      U7 = 1: RETURN
20120  IF  MID$ (A$,Q,1) = "(" THEN
       RETURN
20125  IF  MID$ (A$,Q,1) = "=" THEN
       RETURN
20130  NEXT Q
20140  RETURN
20160 P = RPA(RA):PJ = RUR(1) - 1: RETURN

20170 PJ = RPA(RA):P = EQUAL:PJ = P
      J - 1: RETURN
20180  FOR XY = F2 TO F3 STEP  - 1
20190  GOSUB 20040: IF NA = 0 THEN
       GOTO 20250
20200  GOSUB 20080: IF Q = F3 THEN
      Q = Q + 1
20210 UN =  VAL ( MID$ (A$,Q,D3)):U
      N = UN * MI * MULT: IF UN = 0 THEN
      UN = 1: IF U7 = 1 THEN UN = UN
       * - 1
20230  ON NA GOSUB 1,2,3,4,5,6,7,8,
      9,10,11,12,13,14,15,16,17,18,1
      9,20,21,22,23,24,25,26
20240  GOTO 20290
20250  GOSUB 20080: IF Q = F3 THEN
      Q = Q + 1
```

```
20260 NUM =  VAL ( MID$ (A$,Q,D3)):
      NUM = NUM * MI * MULT: IF X4 =
      1 THEN NUM = NUM *  - 1
20280 JO = JO + NUM
20290  IF Q = F3 THEN  GOTO 20330
20300  IF Q - 1 = F3 THEN  GOTO 203
      30
20310 PJ = Q - 1
20320  NEXT XY
20330  RETURN
20340 PJ = RPA(G8):P = EQUAL:PJ = P
      J - 1: RETURN
20360 G8 = 4
20370  FOR B3 = 1 TO 3
20380 G8 = G8 - 1
20385  IF RPA(G8) = 0 THEN  GOTO 20
      450
20390 Y = RPA(G8) - 1: IF  MID$ (A$
      ,Y,1) = "-" THEN M(G8) =  - 1:
       GOTO 20450
20400 M(G8) = 1
20410  GOSUB 20340: GOSUB 20080: IF
      Q = P THEN Q = Q + 1
20420  IF Q <  > RPA(G8 - 1) THEN  GOTO
      20440
20430 Q = Q + 1
20440 T(G8) =  VAL ( MID$ (A$,Q,D3)
      )
20450  NEXT B3
20460  FOR T5 = 1 TO 3
20470  IF T(T5) = 0 THEN T(T5) = 1
20474  IF M(T5) = 0 THEN M(T5) = 1
```

```
20475  NEXT T5
20480  RETURN
20490 P = RUR(1):PJ = RUR(2) - 1: RETURN

20520 P = RUR(2):PJ = RUR(3) - 1: RETURN

20550  IF AA > 0 THEN  PRINT "+";AA
      ;"A";
20560  IF AA < 0 THEN  PRINT AA;"A"
      ;
20570  IF BB > 0 THEN  PRINT "+";BB
      ;"B";
20580  IF BB < 0 THEN  PRINT BB;"B"
      ;
20590  IF CC > 0 THEN  PRINT "+";CC
      ;"C";
20600  IF CC < 0 THEN  PRINT CC;"C"
      ;
20610  IF DD > 0 THEN  PRINT "+";DD
      ;"D";
20620  IF DD < 0 THEN  PRINT DD;"D"
      ;
20630  IF EE > 0 THEN  PRINT "+";EE
      ;"E";
20640  IF EE < 0 THEN  PRINT EE;"E"
      ;
20650  IF FF > 0 THEN  PRINT "+";FF
      ;"F";
20660  IF FF < 0 THEN  PRINT FF;"F"
      ;
20670  IF GG > 0 THEN  PRINT "+";GG
      ;"G";
```

```
20680  IF GG < 0 THEN  PRINT GG;"G"
       ;
20690  IF HH > 0 THEN  PRINT "+";HH
       ;"H";
20700  IF HH < 0 THEN  PRINT HH;"H"
       ;
20710  IF II > 0 THEN  PRINT "+";II
       ;"I";
20720  IF II < 0 THEN  PRINT II;"I"
       ;
20730  IF JJ > 0 THEN  PRINT "+";JJ
       ;"J";
20740  IF JJ < 0 THEN  PRINT JJ;"J"
       ;
20750  IF KK > 0 THEN  PRINT "+";KK
       ;"K";
20760  IF KK < 0 THEN  PRINT KK;"K"
       ;
20770  IF LL > 0 THEN  PRINT "+";LL
       ;"L";
20780  IF LL < 0 THEN  PRINT LL;"L"
       ;
20790  IF MM > 0 THEN  PRINT "+";MM
       ;"M";
20800  IF MM < 0 THEN  PRINT MM;"M"
       ;
20810  IF NN > 0 THEN  PRINT "+";NN
       ;"N";
20820  IF NN < 0 THEN  PRINT NN;"N"
       ;
20830  IF OO > 0 THEN  PRINT "+";OO
       ;"O";
```

```
20840  IF OO < 0 THEN  PRINT OO;"O"
      ;
20850  IF PP > 0 THEN  PRINT "+";PP
      ;"P";
20860  IF PP < 0 THEN  PRINT PP;"P"
      ;
20870  IF QQ > 0 THEN  PRINT "+";QQ
      ;"Q";
20880  IF QQ < 0 THEN  PRINT QQ;"Q"
      ;
20890  IF RR > 0 THEN  PRINT "+";RR
      ;"R";
20900  IF RR < 0 THEN  PRINT RR;"R"
      ;
20910  IF SS > 0 THEN  PRINT "+";SS
      ;"S";
20920  IF SS < 0 THEN  PRINT SS;"S"
      ;
20930  IF TT > 0 THEN  PRINT "+";TT
      ;"T";
20940  IF TT < 0 THEN  PRINT TT;"T"
      ;
20950  IF UU > 0 THEN  PRINT "+";UU
      ;"U";
20960  IF UU < 0 THEN  PRINT UU;"U"
      ;
20970  IF VV > 0 THEN  PRINT "+";VV
      ;"V";
20980  IF VV < 0 THEN  PRINT VV;"V"
      ;
20990  IF WW > 0 THEN  PRINT "+";WW
      ;"W";
```

```
21000  IF WW < 0 THEN  PRINT WW;"W"
     ;
21010  IF XX > 0 THEN  PRINT "+";XX
     ;"X";
21020  IF XX < 0 THEN  PRINT XX;"X"
     ;
21030  IF YY > 0 THEN  PRINT "+";YY
     ;"Y";
21040  IF YY < 0 THEN  PRINT YY;"Y"
     ;
21050  IF ZZ > 0 THEN  PRINT "+";ZZ
     ;"Z";
21060  IF ZZ < 0 THEN  PRINT ZZ;"Z"
     ;
21070  NORMAL : RETURN
21080  PRINT  LEFT$ (A$,0);: RETURN

21090  IF JO > 0 THEN  PRINT "+";JO
     ;: NORMAL : RETURN
21100  IF JO < 0 THEN  PRINT JO;
21101  NORMAL : RETURN
21110 A9 = AA:B9 = BB:C9 = CC:D9 =
     DD:E9 = EE:F9 = FF:G9 = GG:H9 =
     HH:I9 = II:J9 = JJ:K9 = KK:L9 =
     LL:M9 = MM:N9 = NN:O9 = OO:P9 =
     PP:Q9 = QQ:R9 = RR:S9 = SS:T9 =
     TT:U9 = UU:V9 = VV:W9 = WW:X9 =
     XX:Y9 = YY:Z9 = ZZ
21115 JA = JO:JO = 0
21120 AA = 0:BB = 0:CC = 0:DD = 0:E
     E = 0:FF = 0:GG = 0:HH = 0:II =
     0:JJ = 0:KK = 0:LL = 0:MM = 0:
```

```
     NN = 0:OO = 0:PP = 0:QQ = 0:RR
      = 0:SS = 0:TT = 0:UU = 0:VV =
     0:WW = 0:XX = 0:YY = 0:ZZ = 0:
      RETURN
21130 AA = AA - A9:BB = BB - B9:CC =
     CC - C9:DD = DD - D9:EE = EE -
     E9:FF = FF - F9:GG = GG - G9:H
     H = HH - H9:II = II - I9:JJ =
     JJ - J9:KK = KK - K9:LL = LL -
     L9:MM = MM - M9:NN = NN - N9:O
     O = OO - O9:PP = PP - P9:QQ =
     QQ - Q9:RR = RR - R9:SS = SS -
     S9:TT = TT - T9:UU = UU - U9:V
     V = VV - V9:WW = WW - W9:XX =
     XX - X9:YY = YY - Y9:ZZ = ZZ -
     Z9
21140  RETURN
21150  PRINT : PRINT "-------------
     ---------------------------": CLEAR
     : GOTO 90
21160 AA = AA *  - 1:BB = BB *  - 1
     :CC = CC *  - 1:DD = DD *  - 1
     :EE = EE *  - 1:FF = FF *  - 1
     :GG = GG *  - 1:HH = HH *  - 1
     :II = II *  - 1:JJ = JJ *  - 1
     :KK = KK *  - 1:LL = LL *  - 1
     :MM = MM *  - 1:NN = NN *  - 1
     :OO = OO *  - 1:PP = PP *  - 1
     :QQ = QQ *  - 1:RR = RR *  - 1
     :SS = SS *  - 1:TT = TT *  - 1
     :UU = UU *  - 1:VV = VV *  - 1
     :WW = WW *  - 1:XX = XX *  - 1
```

```
     :YY = YY * - 1:ZZ = ZZ * - 1

21165  RETURN
21170 AA = AA / DV:BB = BB / DV:CC =
      CC / DV:DD = DD / DV:EE = EE /
      DV:FF = FF / DV:GG = GG / DV:H
      H = HH / DV:II = II / DV:JJ =
      JJ / DV:KK = KK / DV:LL = LL /
      DV:MM = MM / DV:NN = NN / DV:O
      O = OO / DV:PP = PP / DV:QQ =
      QQ / DV:RR = RR / DV:SS = SS /
      DV:TT = TT / DV:UU = UU / DV:V
      V = VV / DV:WW = WW / DV:XX =
      XX / DV:YY = YY / DV:ZZ = ZZ /
      DV
21180 JO = JO / DV: RETURN
21190 K2 = 0: IF AA = 0 AND BB = 0 AND
      CC = 0 AND DD = 0 AND EE = 0 AND
      FF = 0 AND GG = 0 AND HH = 0 AND
      II = 0 AND JJ = 0 AND KK = 0 AND
      LL = 0 AND MM = 0 AND NN = 0 AND
      OO = 0 AND PP = 0 AND QQ = 0 AND
      RR = 0 AND SS = 0 AND TT = 0 AND
      UU = 0 AND VV = 0 AND WW = 0 AND
      XX = 0 AND YY = 0 AND ZZ = 0 AND
      JO = 0 THEN K2 = 1
21200  RETURN
21205 K3 = 0: IF AA = 0 AND BB = 0 AND
      CC = 0 AND DD = 0 AND EE = 0 AND
      FF = 0 AND GG = 0 AND HH = 0 AND
      II = 0 AND JJ = 0 AND KK = 0 AND
      LL = 0 AND MM = 0 AND NN = 0 AND
```

```
    OO = O AND PP = O AND QQ = O AND
    RR = O AND SS = O AND TT = O AND
    UU = O AND VV = O AND WW = O AND
    XX = O AND YY = O AND ZZ = O THEN
    K3 = 1
21206  RETURN
21210  GOSUB 21190: IF K2 = 1 THEN
     PRINT "+O";
21220  RETURN
21230  IF JA = O THEN  PRINT "+O";
21240  RETURN
21750  IF JA > O THEN  PRINT "+";JA
    ;
21760  IF JA < O THEN  PRINT JA;
21770  NORMAL : RETURN
22000 LIN$ = " *** *** *** *** ***
    *** *   * *   * *** * * * * *
    * *   * * * * ** ** ** ** * *
    "
22010 LIN$ = LIN$ + "*** *** * * *
    * *** *** * * * * * * *** *
    **  * * * * **  * * *   * *
    * * * "
22020 LIN$ = LIN$ + "*   * * *** **
    * * * * * *   * *   * * *"
22030 NIL$ = "    *** *** *** *** *
       * *** *** ***       *   * *
     *    *  *   * * * * * *
       "
22040 NIL$ = NIL$ + "*** * * **   *
      * * * *** *** **          *
    * * *    *  ** ** * * **  *
```

```
                  "
22050 NIL$ = NIL$ + "*** *** *     *
       *   * * * * * ***   "
22060  SPEED= 255: NORMAL : TEXT : HOME
      : VTAB 2
22070  FOR I = 1 TO 200:A$ =  MID$
      (LIN$,I,1): IF A$ = "*" THEN  PRINT
      " ";: GOTO 22090
22080  NORMAL : PRINT " ";: INVERSE

22090  NEXT I: PRINT : PRINT : PRINT

22100  FOR I = 1 TO 200:A$ =  MID$
      (NIL$,I,1): IF A$ = "*" THEN  PRINT
      " ";: GOTO 22120
22110  NORMAL : PRINT " ";: INVERSE

22120  NEXT I: NORMAL
22130 LIN$ = "ALGEBRA":NIL$ = "JOSE
       A. FARRAN"
22140  VTAB 17: HTAB 21 -  LEN (LIN
      $) / 2: PRINT LIN$: PRINT : HTAB
      21 -  LEN (NIL$) / 2: PRINT NI
      L$: PRINT
22150  HTAB 10: PRINT "PROGRAMMA IN
      TERNATIONAL": PRINT : HTAB 12:
       PRINT "COPYRIGHT (C) 1981"
22160  WAIT  - 16384,128: POKE  - 1
      6368,0: RETURN
40000  PRINT "YOU HAVE EXCEADED THR
      EE '(' PARENTASIS": PRINT "ON
      RIGHT SIDE OF EQUATION": END
```

```
40010  PRINT "YOU HAVE EXCEADED THR
     EE ')' PARENTASIS": PRINT "ON
     RIGHT SIDE OF EQUATION": END
40020  TEXT : HOME : SPEED= 170: VTAB
     5: PRINT  TAB( 10)"----------
     --------": PRINT  TAB( 13)"A L
      G E B R A": PRINT  TAB( 13)"-
     -----------": PRINT
40025  PRINT " 1'ST DEGREE EQUATION
      SOLVING PROGRAM": PRINT : PRINT
      TAB( 18)"BY": PRINT : PRINT  TAB(
     12)"JOSE A. FARRAN": PRINT : PRINT
      TAB( 14)"APRIL 1979": SPEED=
     255: GOTO 85
40030  HOME : PRINT "**************
     **************************"
40040  PRINT : PRINT "VARIABLES:-":
      PRINT "----------": PRINT "UP
     TO '26' VARIABLES MAY BE USED.
     ": PRINT : PRINT "VALUES MAY B
     E ASSINGED, FOR EXAMPLE": PRINT
     : PRINT "'10A','-2X','Y', A VA
     RIABLE WITHOUT A": PRINT : PRINT
     "VALUE AS IN 'Y', WILL BE CONS
     IDER AS": PRINT : PRINT "'1Y',
      ALSO '-Y' IS '-1Y'."
40050  PRINT : PRINT "NUMBERS ALONE
      ARE PERMITED, THEY MAY BE": PRINT
     : PRINT "TOGETHERE WITH VARIAB
     LES SEPERATED BY": PRINT : PRINT
     "SIGNS, AS '2X+2' THE COMPUTER
      WILL THEN": PRINT : PRINT "AD
```

```
      D ALL ALIKE VARIABLES & NUMBER
      S, AND": PRINT : PRINT "PRINT
      OUT THE RESULT"
40070 GOSUB 49000: PRINT "EQUAL SI
      GN:-": PRINT "----------"
40075 PRINT "AN EQUAL SIGN MAY BE
      ENTERED WITH YOUR": PRINT : PRINT
      "EQUATION, THE COMPUTER WILL T
      HEN SOLVE": PRINT : PRINT "THE
       PROBLEM, AND EVENTUALLY BRING
       ALL": PRINT : PRINT "THE VARI
      ABLES TO THE LEFT & NUMBERS TO
      ": PRINT : PRINT "THE RIGHT SI
      DE OF THE EQUAL SIGN."
40080 PRINT : PRINT "THE SIGN (+,-
      ) OF A NUMBER OR VARIABLE": PRINT
      : PRINT "THAT IS TAKEN OVER TH
      E EQUAL SIGN WILL": PRINT : PRINT
      "BE CHANGED TO IT'S OPPOSITE":
       PRINT : PRINT "AUTOMATICALLY.
       IF AN EQUAL SIGN IS": PRINT :
       PRINT "ENTERED, PLEASE DON'T
      LEAVE ANY SIDE": PRINT
40090 FLASH : PRINT "BLANK.": NORMAL
      : GOSUB 49000
40100 PRINT "SOLVING FOR:-": PRINT
      "-----------"
40110 PRINT "YOU MAY SOLVE FOR ANY
       ONE VARIABLE IF": PRINT : PRINT
      "AN EQUAL SIGN WAS ENTERED, TH
       E VARIABLE": PRINT : PRINT "YO
       UR SOLVING FOR WILL THEN BE LE
```

```
FT": PRINT : PRINT "ON THE LEF
T SIDE, AND THE REST OF THE": PRINT
: PRINT "VARIABLES & NUMBERS W
ILL BE TAKEN TO": PRINT
40120 PRINT "THE RIGHT SIDE OF THE
 EQUAL SIGN.": PRINT : PRINT "
THE VALUE OF THE VARIABLE WILL
 THEN BE": PRINT : PRINT "DIVI
DED BY THE REST OF THE VARIABL
ES &": PRINT : PRINT "NUMBERS.
 FOR EXAMPLE IF YOU ENTERED": PRINT
: PRINT "'2X=10' AND YOUR SOLV
ING FOR 'X' THEN": PRINT
40130 PRINT "THE FINAL ANSWER WILL
 BE 'X=5'.": GOSUB 49000
40140 INVERSE : PRINT "INVERSE";: NORMAL
: PRINT ":-": PRINT "--------":
 PRINT
40150 PRINT "NUMBERS & VARIABLES T
HAT APEAR IN": PRINT : PRINT "
INVERSE MEANS THAT THE NUMER'S
' IS": PRINT : PRINT "BEING SU
BTRACTED FROM EACH SIDE OF THE
": PRINT : PRINT "EQUAL SIGN,
IN ORDER TO BRING THE": PRINT
: PRINT "NUMBER OVER THE EQUAL
 SIGN.": PRINT
40160 PRINT "THE SIGN OF THE INVER
SE NUMBER WILL BE": PRINT : PRINT
"THE OPPOSITE OF THE NUMBER TO
 BE": PRINT : PRINT "SUBTRACTE
D FORM EACH SIDE, FOR EXAMPLE-
```

```
      ": PRINT
40170  PRINT "IF YOU ENTER-
       2X+4A=10+4A": PRINT "
                  2X+4A";: INVERSE : PRINT
      "-4A";: NORMAL : PRINT "=10+4A
      ";: INVERSE : PRINT "-4A": NORMAL
      : PRINT "FINAL ANSWER IS--
         2X=10": PRINT
40175  GOSUB 49000
40180  PRINT "MULTIPLICATION:-
                  PAGE 1": PRINT "--
      -------------"
40190  PRINT "SINCE THIS IS A FIRST
       DEGREE EQUATION": PRINT : PRINT
      "SOLVING PROGRAM, THE ONLY": PRINT
      : PRINT "MULTIPLICATION ALLOWE
      D IS NUMBERS TIMES": PRINT : PRINT
      "NUMBERS & VARIABLES ONLY, THE
       ONLY WAY": PRINT : PRINT "TO
      MULTIPLY IS WITH A NUMBER AND
      A SET": PRINT
40200  PRINT "OF PARENTASIS AFTER I
      T, UP TO THREE SET": PRINT : PRINT
      "OF PARENTASIS '((()))' MAY BE
       INCLUDED,": PRINT : PRINT "YO
      U MAY ENTER ANY SET OF PARENTA
      SIS IF": PRINT : PRINT "YOU EN
      TERED AN EQUAL SIGN AND THE": PRINT
      : PRINT "PARENTASIS ARE ON THE
       RIGHT SIDE OF THE": PRINT : PRINT
      "EQUAL SIGN."
40205  GOSUB 49000: PRINT  TAB( 33)
```

```
"PAGE 2": PRINT
40210  PRINT "PARENTASIS HAVE TO BE
 IN THE FOLLOWING": PRINT : PRINT
"ORDER '()', OR '(())', OR '((
()))'.": PRINT : PRINT "ANY OR
DER LIKE THE FOLLOWING IS NOT"
: PRINT : PRINT "ALLOWED '()()
'.": PRINT : PRINT "LET'S SAY
YOU WAMT TO MULTIPLY '6'": PRINT
: PRINT "TIMES '6+6A'. YOU ENT
ER THE FOLLOWING": PRINT
40220  PRINT "'0=6(6+6A)'. OR IF YO
U WANT TO KNOW": PRINT : PRINT
"WHAT IS THE NEGATIVE OF '6+6A
' YOU ENTER": PRINT "'0=-(6+6A
)'. IN A EQUATION LIKE": PRINT
: PRINT "'2X=2(3(4(2A)-2S))' T
HERE CAN ONLY BE A": PRINT : PRINT
"NUMBER IN FRONT OF THE PARENT
ASIS, WHEN"
40225  GOSUB 49000: PRINT  TAB( 33)
"PAGE 3": PRINT
40230  PRINT "CLOSING THE PARENTASI
S THEN YOU MAY HAVE": PRINT "V
ARIOUS NUMBER'S & VARIABLES, F
OR FUTHER": PRINT "EXAMPLES LO
OK UNDER EQUATION EXAMPLES:-":
 PRINT : PRINT "WHEN MULTIPLIN
G THE FOLLOWING ARE NOT": PRINT
: PRINT "ALLOWED IN FRONT OF T
HE PARENTASIS-": PRINT
40240  PRINT "'1','0','+'. ONE TIME
```

```
      S ANYTHING IS ONE,": PRINT : PRINT
      "ZERO WILL RESULT IN O NUMBER,
       AND PLUS": PRINT : PRINT "SIG
      N DOSEN'T CHANGE ANYTHING."
40250  PRINT : PRINT "ALSO ANY VARI
      ABLE THAT IS INSIDE A": PRINT
      : PRINT "PARENTASIS, AND IS AL
      ONE AS IN 'Y', IT": PRINT : PRINT
      "MUST BE ENTER AS '1Y'.
40255  GOSUB 49000: PRINT  TAB( 33)
      "PAGE 4": PRINT
40260  PRINT "FOR EXAMPLE:-": PRINT
      : PRINT "IF YOU ENTER
               5X=2(Y)": PRINT : PRINT
      "THE 'Y' MUST BE": PRINT : PRINT
      "CHANGE TO
       5X=2(1Y)": PRINT "-----------
      ----------------------------"
40270  PRINT "FOR EASIER READING OF
       YOUR EQUATION": PRINT : PRINT
      "DON'T MAKE YOUR EQUATION LONG
      ER THAN": PRINT : PRINT "35 CH
      ARACTERS LONG."
40280  GOSUB 49000: PRINT "EQUATION
       EXAMPLES:-           LAST PAG
      E": PRINT "-----------------"
40290  PRINT  TAB( 5)"ADDITION OF N
      UMBERS & VARIABLES": PRINT
40300  PRINT "5+4F-100       10X+10Y
      +X+Y      -2A-1A-A": PRINT : PRINT
      "5U+10G+5U      90-2R-100-R
       -2P+2W+2"
```

```
40310  PRINT "------------------------
     -------------------": PRINT : PRINT
     "EQUATIONS WITH '=' SIGN, SOLV
     ING FOR 'X'": PRINT "20=5X+10
         10A=10B-5X        R+X=-Z-T":
      PRINT : PRINT "25X=1000      -
     X=10Z-50Y        2X=10+10"
40320  PRINT "------------------------
     -------------------": PRINT  TAB(
     10)"COMPLEX EQUATIONS": PRINT

40330  PRINT "10+2A=100-(2(8(2A-10)
     -10)-10A)-100+20X": PRINT : PRINT
     "90+90A=10-20X+2(3(4(2A)-10A))
     -100+15C-Z": PRINT : PRINT "20
     X-15S=20X-2(3A)-100+20S+100+90
     C-115X"
40340  GOSUB 49000: RETURN
49000  VTAB (24): PRINT  TAB( 24)"(
     HIT ANY KEY) ";: GET Z$: HOME
     :CO = 0: RETURN
49910  REM
```

程式十一　電話號碼簿

由電腦來處理像資料記錄、歸納、存檔等繁複而單調的工作是再恰當不過的了，這裡所介紹的一個程式，便是如何利用電腦來做像電話號碼記錄、尋找之類的工作，雖然只是資料處理的最簡單應用，可是如果把家庭裡的雜務也類似的全部放到一個程式裡頭來處理，或是把它加以發展，則未嘗不能構造出巧妙的應用實例的。

本程式總共能容納150個電話號碼及人名，並且可以執行①搜尋一個號碼，②列出全部的名單，③填入新的名單，④消去原有名單，⑤修改原有名單，⑥叫印字機印出資料，⑦將資料存錄至磁碟保存。

程　式

```
]LIST

100  REM ***********************
101  REM *                     *
102  REM *    W A R N I N G    *
103  REM *    - - - - - - -    *
104  REM *                     *
105  REM *    DO NOT MODIFY    *
106  REM *  ANY LINE IN THIS   *
107  REM *   PROGRAM WITH A    *
108  REM *  LINE NUMBER LESS   *
```

```
109  REM *      THAN 400!!       *
110  REM *                       *
111  REM *    PROGRAM WILL       *
112  REM *   SELF-DESTRUCT!      *
113  REM *                       *
114  REM ************************
115  REM


200  DATA 1001
201  DATA APPLE COMPUTER INC.......
     ...408 996 1010
202  DATA \........................
     ...............
203  DATA \........................
     ...............
204  DATA \........................
     ...............
205  DATA \........................
     ...............
206  DATA \........................
     ...............
207  DATA \........................
     ...............
208  DATA \........................
     ...............
209  DATA \........................
     ...............
210  DATA \........................
     ...............
211  DATA \........................
     ...............
```

```
212  DATA \........................
     ...............
213  DATA \........................
     ...............
214  DATA \........................
     ...............
215  DATA \........................
     ...............
216  DATA \........................
     ...............
217  DATA \........................
     ...............
218  DATA \........................
     ...............
219  DATA \........................
     ...............
220  DATA \........................
     ...............
221  DATA \........................
     ...............
222  DATA \........................
     ...............
223  DATA \........................
     ...............
224  DATA \........................
     ...............
225  DATA \........................
     ...............
226  DATA \........................
     ...............
227  DATA \........................
     ...............
```

```
228  DATA \........................
     ...............
229  DATA \........................
     ...............
230  DATA \........................
     ...............
231  DATA \........................
     ...............
232  DATA \........................
     ...............
233  DATA \........................
     ...............
234  DATA \........................
     ...............
235  DATA \........................
     ...............
236  DATA \........................
     ...............
237  DATA \........................
     ...............
238  DATA \........................
     ...............
239  DATA \........................
     ...............
240  DATA \........................
     ...............
241  DATA \........................
     ...............
242  DATA \........................
     ...............
243  DATA \........................
     ...............
```

```
244  DATA \........................
     ...............
245  DATA \........................
     ...............
246  DATA \........................
     ...............
247  DATA \........................
     ...............
248  DATA \........................
     ...............
249  DATA \........................
     ...............
250  DATA \........................
     ...............
251  DATA \........................
     ...............
252  DATA \........................
     ...............
253  DATA \........................
     ...............
254  DATA \........................
     ...............
255  DATA \........................
     ...............
256  DATA \........................
     ...............
257  DATA \........................
     ...............
258  DATA \........................
     ...............
259  DATA \........................
     ...............
```

```
260  DATA \........................
     ...............
261  DATA \........................
     ...............
262  DATA \........................
     ...............
263  DATA \........................
     ...............
264  DATA \........................
     ...............
265  DATA \........................
     ...............
266  DATA \........................
     ...............
267  DATA \........................
     ...............
268  DATA \........................
     ...............
269  DATA \........................
     ...............
270  DATA \........................
     ...............
271  DATA \........................
     ...............
272  DATA \........................
     ...............
273  DATA \........................
     ...............
274  DATA \........................
     ...............
275  DATA \........................
     ...............
```

```
276  DATA \........................
     ...............
277  DATA \........................
     ...............
278  DATA \........................
     ...............
279  DATA \........................
     ...............
280  DATA \........................
     ...............
281  DATA \........................
     ...............
282  DATA \........................
     ...............
283  DATA \........................
     ...............
284  DATA \........................
     ...............
285  DATA \........................
     ...............
286  DATA \........................
     ...............
287  DATA \........................
     ...............
288  DATA \........................
     ...............
289  DATA \........................
     ...............
290  DATA \........................
     ...............
291  DATA \........................
     ...............
```

```
292  DATA \........................
     ...............
293  DATA \........................
     ...............
294  DATA \........................
     ...............
295  DATA \........................
     ...............
296  DATA \........................
     ...............
297  DATA \........................
     ...............
298  DATA \........................
     ...............
299  DATA \........................
     ...............
300  DATA \........................
     ...............
301  DATA \........................
     ...............
302  DATA \........................
     ...............
303  DATA \........................
     ...............
304  DATA \........................
     ...............
305  DATA \........................
     ...............
306  DATA \........................
     ...............
307  DATA \........................
     ...............
```

```
308  DATA \........................
     ...............
309  DATA \........................
     ...............
310  DATA \........................
     ...............
311  DATA \........................
     ...............
312  DATA \........................
     ...............
313  DATA \........................
     ...............
314  DATA \........................
     ...............
315  DATA \........................
     ...............
316  DATA \........................
     ...............
317  DATA \........................
     ...............
318  DATA \........................
     ...............
319  DATA \........................
     ...............
320  DATA \........................
     ...............
321  DATA \........................
     ...............
322  DATA \........................
     ...............
323  DATA \........................
     ...............
```

```
324  DATA \........................
     ...............
325  DATA \........................
     ...............
326  DATA \........................
     ...............
327  DATA \........................
     ...............
328  DATA \........................
     ...............
329  DATA \........................
     ...............
330  DATA \........................
     ...............
331  DATA \........................
     ...............
332  DATA \........................
     ...............
333  DATA \........................
     ...............
334  DATA \........................
     ...............
335  DATA \........................
     ...............
336  DATA \........................
     ...............
337  DATA \........................
     ...............
338  DATA \........................
     ...............
339  DATA \........................
     ...............
```

```
340  DATA \........................
     ...............
341  DATA \........................
     ...............
342  DATA \........................
     ...............
343  DATA \........................
     ...............
344  DATA \........................
     ...............
345  DATA \........................
     ...............
346  DATA \........................
     ...............
347  DATA \........................
     ...............
348  DATA \........................
     ...............
349  DATA \........................
     ...............
350  DATA \........................
     ...............
400  REM

SET PRINTER SLOT


410  LET SL = 1
450  REM

DISK SYSTEM??
```

```
460  IF  PEEK (994) +  PEEK (1001) =
     192 THEN DISK = 1
470  IF DISK THEN  PRINT  CHR$ (4)"
     NOMON C,I,O": HOME


500  REM   GET ADRESS OF FIRST DATA
      STATEMENT
510  LET START = ( PEEK (103) +  PEEK
     (104) * 256 + 458)
550  REM   SET FAST/SLOW SEARCH
560  LET OP = 0
570  GOTO 25000
700  REM   ACTIVATE PRINTER
710  PR# SL: RETURN
750  REM  DE-ACTIVATE PRINTER
760  PR# 0: RETURN
800  REM   END  ROUTINE
810  IF MF = 0 THEN 890
820  IF DISK THEN  PRINT  CHR$ (4)"
     SAVE PHONE LIST": TEXT : HOME
     : VTAB 22: CALL 976: END
830  HOME : VTAB 11: PRINT "INSERT
     A NEW TAPE INTO THE RECORDER,
     SET";: PRINT "ON RECORD, AND P
     RESS 'RETURN' ";: INPUT "";S$
840  FOR I = 1 TO 1500: NEXT : SAVE
     : FOR I = 1 TO 2000: NEXT : SAVE

850  HOME : VTAB 12: PRINT "THE PRO
     GRAM AND DATA HAVE BEEN SAVED!
     !": VTAB 22: END
```

```
890  TEXT : HOME : VTAB 22: END
1000  REM

SEARCH


1005  GOSUB 3000: IF OP = 1 THEN 10
     10
1006  HOME : VTAB 9: HTAB 5: PRINT
     "WOULD YOU LIKE TO SEARCH BY":
      PRINT : HTAB 10: PRINT "1  WO
     RD (FAST)": HTAB 7: PRINT "OR
     2  CHARACTER (SLOW)"
1007  VTAB 14: CALL  - 868: HTAB 20
     : INPUT "WHICH: ";SK$:SK =  VAL
     (SK$): IF SK <  > 1 AND SK <  >
     2 OR  LEN (SK$) > 1 THEN  PRINT
     "": GOTO 1007
1010  HOME : VTAB 11: HTAB 5: PRINT
     "PLEASE ENTER THE SEARCH KEY:"
     : VTAB 13: HTAB 5: INPUT "";S$
     : IF S$ = "" THEN 25000
1012  IF PR THEN  GOSUB 700: PRINT
     " ": PRINT " "
1015  HOME : VTAB 1: PRINT " NAME
                          PHONE NUMB
     ER": PRINT "------------------
     -----------------------"
1017  IF PR THEN  GOSUB 750
1018  IF S = 2 THEN 2510
1020  VTAB 12: CALL  - 868: HTAB 12
     : INVERSE : PRINT "<< SEARCHIN
```

```
      G >>": NORMAL :VT = 2
1025  RESTORE : READ Z
1030  LET FO = 0: FOR I = 1 TO NR
1040  READ NA$: IF  LEFT$ (NA$,1) =
      CHR$ (92) THEN 1040
1045  IF SK = 0 OR SK = 1 THEN  IF
      LEFT$ (NA$, LEN (S$)) <  > S$
      THEN 1075
1050  FOR II = 1 TO  LEN (NA$)
1060  IF  MID$ (NA$,II, LEN (S$)) =
      S$ THEN FO = FO + 1: GOSUB 110
      0: GOTO 1075
1070  NEXT II
1075  NEXT I: IF FO THEN 1200
1080  VTAB 12: CALL  - 868: HTAB 12
      : INVERSE : PRINT "<< NOT FOUN
      D >>": NORMAL : IF PR THEN  GOSUB
      700: FOR S = 1 TO 4: PRINT " "
      : NEXT : GOSUB 750
1090  PRINT  CHR$ (7): FOR I = 1 TO
      1500: NEXT I: GOTO 25000
1099  STOP
1100  REM

GOT ONE!


1102  LET VT = VT + 1: IF VT >  = 2
      1 AND PR <  > 1 THEN  GOSUB 13
      00
1104  IF FO = 1 THEN  VTAB 12: CALL
       - 868
```

```
1105  IF PR THEN  GOSUB 700
1110  VTAB VT: HTAB 1: PRINT NA$;
1120  IF PR THEN  GOSUB 750
1140  IF I <  = NR THEN  RETURN
1200  IF PR THEN  GOSUB 700: PRINT
     " ": PRINT " ": PRINT " ": PRINT
     " ": GOSUB 750
1210  VTAB 22: CALL  - 868: PRINT "
     THAT'S ALL OF THEM! - PRESS 'R
     ETURN' ";: INPUT "";S$: GOTO 2
     5000
1299  STOP
1300  REM

GOT A FULL PAGE



1310  VTAB 22: CALL  - 958: PRINT "
     ENTER 'RTN' TO CONT - 'ESC' FO
     R MENU ";: GET T$
1320  IF T$ =  CHR$ (27) THEN  POP
     : GOTO 25000
1330  IF T$ =  CHR$ (13) THEN  VTAB
     3: HTAB 1: CALL  - 958:VT = 3:
      RETURN
1340  PRINT  CHR$ (7): GOTO 1300
2499  STOP
2500  REM

LIST EVERYTHING
```

```
2504  GOSUB 3000
2505  GOTO 1012
2510  RESTORE : READ Z
2530  LET VT = 2:FO = 0: FOR I = 1 TO
     NR
2540  READ NA$: IF  LEFT$ (NA$,1) =
      CHR$ (92) AND I <  = NR THEN
     2540
2542  GOSUB 1100
2545  IF VT >  = 22 THEN VT = 2: VTAB
     3: CALL  - 958
2550  NEXT I
2560  GOTO 1200
3000  REM

IS DATA FILE EMPTY?


3010  HOME : RESTORE : IF NR = 0 THEN
      VTAB 12: HTAB 9: INVERSE : PRINT
     "<< NO NAMES ON FILE  >>": NORMAL
     : FOR I = 1 TO 2000: NEXT I: POP
     : GOTO 25000
3020  RETURN
4000  REM

ENTER LISTINGS


4010  HOME : IF NR >  = 150 THEN  VTAB
     12: HTAB 12: PRINT "DATA TABLE
      FULL!": FOR I = 1 TO 2000: NEXT
```

```
     : GOTO 25000
4020  POKE 32,3: VTAB 10: CALL  - 9
     58: PRINT "ENTER NEW NAME AND
     PRESS 'RETURN'.": PRINT "(USE
     NO COMMAS, COLONS OR QUOTES.":
      PRINT "TWENTY-FIVE CHARACTERS
     , MAXIMUM.)"
4030  VTAB 14: INPUT "-> ";A$: GOSUB
     6100:NN$ = B$: IF NN$ = "" AND
     S = 5 THEN  POKE 32,0: POP : GOTO
     25000
4035  IF NN$ = "" THEN  POKE 32,0: GOTO
     25000
4040  IF  LEN (NN$) > 25 THEN  PRINT
     "": GOTO 4020
4050  VTAB 10: CALL  - 958: PRINT "
     ENTER NEW NUMBER THEN PRESS 'R
     ETURN'.": PRINT "(USE THE FORM
     AT: XXX XXX XXXX)"
4060  VTAB 13: INPUT "-> ";A$: GOSUB
     6100:NP$ = B$: IF NP$ = "" AND
     S = 5 THEN  POKE 32,0: POP : GOTO
     25000
4065  IF NP$ = "" THEN  POKE 32,0: GOTO
     25000
4070  IF  LEN (NP$) <  > 12 THEN  PRINT
     "": GOTO 4050
4100  REM

IS ENTRY CORRECT?
```

```
4110  TEXT : HOME : VTAB 8: PRINT "
     HERE IS THE ";: INVERSE : PRINT
     "NAME";: NORMAL : PRINT " ENTR
     Y:": VTAB 10: HTAB 14: PRINT N
     N$
4120  VTAB 12: PRINT "AND THE ";: INVERSE
     : PRINT "PHONE NUMBER";: NORMAL
     : PRINT ":": VTAB 14: HTAB 14:
      PRINT NP$
4140  VTAB 18: HTAB 8: CALL  - 868:
      INPUT "ARE THEY CORRECT (Y/N)
      ";S$: IF  LEFT$ (S$,1) = "Y" THEN
     4180
4150  IF  LEFT$ (S$,1) <  > "Y" AND
     S = 5 THEN  POP : GOTO 6000
4160  IF  LEFT$ (S$,1) <  > "N" THEN
      PRINT "": GOTO 4100
4170  GOTO 4000
4175  IF S = 5 THEN 4200
4180  IF S = 5 THEN 4200
4190  LET UD = 1
4200  REM

PAD TO 40 CHARACTERS


4210  FOR I = LEN (NN$) TO (39 -  LEN
     (NP$)):NN$ = NN$ + ".": NEXT
4220  LET NN$ = NN$ + NP$: IF S = 5
      THEN  RETURN
4250  REM
```

UPDATE DATA NUMBER

```
4260  RESTORE : READ T:T = T + UD:C
     U = (START - 11)
4270  LET T$ =  STR$ (T): FOR I = 1
      TO  LEN (T$)
4280  POKE (CU + I), ASC ( MID$ (T$
     ,I,1)): NEXT
4290  IF S = 4 THEN  RETURN
4300  REM
```

FIND EMPTY DATA LINE

```
4310  READ T$: IF  LEFT$ (T$,1) <  >
      CHR$ (92) THEN 4310
4320  LET LN =  PEEK (123) +  PEEK
     (124) * 256
4350  REM
```

CALCULATE OFFSET

```
4360  LET CU = (START + ((LN - 201)
      * 46))
4400  REM
```

POKE DATA LINE

```
4410  FOR I = CU TO (CU + 39)
```

```
4420  POKE I, ASC ( MID$ (NN$,I + 1
      - CU,1)): NEXT
4450  REM

SET UPDATE FLAG


4460  LET MF = MF + 1: IF S = 4 OR
     S = 5 THEN  RETURN
4470  GOTO 25000
4499  STOP
5000  REM

DELETE DATA CONTENTS


5005  HOME : GOSUB 3000
5010  HOME : VTAB 10: PRINT "PLEASE
      ENTER THE NAME OF THE PARTY Y
     OU": PRINT "WISH DELETED FROM
     THE DIRECTORY AND": PRINT "PRE
     SS 'RETURN'."
5020  VTAB 14: INPUT "-> ";K$: IF K
     $ = "" AND S = 5 THEN  POP : GOTO
     25000
5030  IF K$ = "" THEN 25000
5050  REM

FIND AN EXACT MATCH


5060  RESTORE : READ Z: FOR I = 1 TO
```

```
     NR
5070  READ NA$: IF  LEFT$ (NA$, LEN
     (K$)) = K$ THEN 5100
5075  IF  LEFT$ (NA$,1) =  CHR$ (92
     ) AND I <  = NR THEN 5070
5080  NEXT : HOME : GOTO 25000
5100  REM

FOUND A MATCH


5110  LET LN =  PEEK (123) +  PEEK
     (124) * 256
5120  HOME : VTAB 11: PRINT NA$: IF
     S = 5 THEN  RETURN
5130  VTAB 13: CALL  - 868: INPUT "
     DO YOU WISH TO DELETE THIS DAT
     A (Y/N) ";S$: IF  LEFT$ (S$,1)
      = "Y" THEN 5200
5140  IF  LEFT$ (S$,1) <  > "N" THEN
      PRINT "": GOTO 5130
5150  GOTO 5080
5200  REM

OK THEN, KILL IT!


5210  LET UD =  - 1: GOSUB 4250
5220  LET CU = (START + ((LN - 201)
      * 46))
5230  POKE CU, ASC ( CHR$ (92))
5240  GOSUB 4450: GOTO 25000
```

```
6000  REM

EDIT A DATA LINE


6010  HOME : GOSUB 3000
6012  HOME : VTAB 10: PRINT "PLEASE
       ENTER THE NAME OF THE PARTY W
      HOSE";: PRINT "LISTING YOU WIS
      H TO EDIT AND PRESS": PRINT "'
      RETURN'.": GOSUB 5020
6020  VTAB 13: CALL  - 868: INPUT "
      DO YOU WISH TO EDIT THIS DATA
      (Y/N) ";S$: IF S$ = "" THEN 25
      000
6030  IF  LEFT$ (S$,1) = "Y" THEN N
      N$ = NA$: GOTO 6050
6035  IF  LEFT$ (S$,1) = "N" THEN  GOSUB
      5070
6040  PRINT "": GOTO 6020
6050  REM

GO EDIT THE LINE


6060  HOME : VTAB 8: PRINT NN$: GOSUB
      4020: GOSUB 4360
6070  GOTO 25000
6100  REM

REMOVE CONTROL CHARS.
```

```
6105 B$ = ""
6107 IF NOT LEN (A$) THEN RETURN
6110 FOR I = 1 TO LEN (A$)
6120 IF ASC ( MID$ (A$,I,1)) > 26
     THEN B$ = B$ + MID$ (A$,I,1)

6130 NEXT I
6140 RETURN
24000 REM PRINTER ON/OFF
24010 LET PR = NOT PR: GOTO 25050

24020 STOP
25000 REM MAIN MENU
25005 RESTORE : READ NR:NR = NR -
     1000
25010 TEXT : HOME : VTAB 1: PRINT
     "________________________________________
     ______________";: INVERSE : REM
     40'_'
25020 PRINT " APPLE COMPUTER, INC.
            PHONE LIST ";: NORMAL
     : PRINT "______________________
     ____________________"
25030 PRINT "PROGRAM CURRENTLY CON
     TAINS "NR" LISTING";: IF NR <
     > 1 THEN PRINT "S";
25040 PRINT : PRINT "ROOM FOR "(15
     0 - NR)" ADDITIONAL LISTINGS"
25050 VTAB 8: PRINT "THE PRINTER I
     S";: IF NOT PR THEN PRINT "
     OFF";
25055 IF NOT PR THEN VTAB 9: PRINT
```

```
             ": GOTO 25070
25060  PRINT " ";: FLASH : PRINT "
     ON ": NORMAL
25070  VTAB 10: PRINT "YOU MAY CHOO
     SE FROM THE FOLLOWING:": POKE
     32,5: PRINT : PRINT
25080  PRINT "1  SEARCH FOR A LISTI
     NG": PRINT "2  LIST THE ENTIRE
      FILE"
25090  PRINT "3  ENTER NEW LISTINGS
     ": PRINT "4  DELETE A LISTING"

25100  PRINT "5  EDIT A LISTING": PRINT
     "6  PRINTER ON/OFF"
25110  PRINT "7  EXIT THIS PROGRAM"
     : POKE 32,0: PRINT
25120  POKE 32,0: PRINT
25130  VTAB 22: HTAB 20: CALL  - 86
     8: INPUT "WHICH: ";S$:S =  VAL
     (S$)
25140  IF S < 1 OR S > 7 THEN  PRINT
      CHR$ (7): GOTO 25130
25150  ON S GOTO 1000,2500,4000,500
     0,6000,24000,800
25160  STOP
65533  REM
```

程式說明

將程式 RUN 之後，跳過一連串151行的DATA，然後在第410行，460行及470行對磁碟機（DISK）及印字機（PRINTER）做啟動工作

，其中 CHR$（ 4 ）是“ CONTROL—D ”，而103及104兩個位址是程式起始位址（通常是 $0801 ）。之後由570行跳至 25000 行將程式的服務項目（MENU）印出，並詢問操作者將所選的項目輸入，而程式就由第 25150 行的ON S GOTO跳至各該對應的服務程式。

由於服務項目總共有 7 項，在此我們謹就第一項爲例子來說明。第一項的功能是由一些已存錄的許多資料中尋找出所希望的名單的電話號碼，程式首先至第 3000 行的副程式將DATA存回（RESTORE），並且檢查資料是否空白，若是則印出<< NO NAME ON FILE >>並且利用 POP 跳出副程式而回到 25000 行的主程式，若不是空白則回到第 1006 行並要求操作者選擇是以字（快）或字元（慢）來尋找，接着由第 1010 行將所要求的名稱以 INPUT S$送入電腦，尋找的工作是由第 1025 行至 1075 行來完成，如果所尋找的名字不在名單裡，則由 1080 行印出<< NOT FOUND >>以及 1090 的 CHR$（ 7 ）（ 即 CONTROL－G ）由喇叭發出響聲並回到 25000 行的出程式。

執行例

```
]RUN
------------------------------------------------------------
 APPLE COMPUTER, INC.          PHONE LIST
------------------------------------------------------------
PROGRAM CURRENTLY CONTAINS 1 LISTING
ROOM FOR 149 ADDITIONAL LISTINGS
THE PRINTER IS  OFF
YOU MAY CHOOSE FROM THE FOLLOWING:


1   SEARCH FOR A LISTING
2   LIST THE ENTIRE FILE
3   ENTER NEW LISTINGS
4   DELETE A LISTING
5   EDIT A LISTING
6   PRINTER ON/OFF
7   EXIT THIS PROGRAM
```

當程式執行時，在螢幕印出 1 至 7 的MENU出來，並詢問選擇項目（WHI（H），如果回答 1（ 即要求尋找某一名單），則出現下列字幕：

```
                 WHICH: 1
WOULD YOU LIKE TO SEARCH BY

     1  WORD (FAST)
  OR 2  CHARACTER (SLOW)
                 WHICH: 2
PLEASE ENTER THE SEARCH KEY:
```

接着就可以把所要找的名字鍵入。當尋找完畢後螢幕又把MENU列出，此時如果鍵入2，則螢幕就把所有的名單列出，在印字機上也會打出所有的名單（如果是使用MX-80等EPSON的印字機，則必須多插入一個PRINT指令，或是印出格式多空一格，或是用POKE 1657，80將輸出交給印字機，否則輸出第一個字是出不來的）：

```
YOU MAY CHOOSE FROM THE FOLLOWING:


1  SEARCH FOR A LISTING
2  LIST THE ENTIRE FILE
3  ENTER NEW LISTINGS
4  DELETE A LISTING
5  EDIT A LISTING
6  PRINTER ON/OFF
7  EXIT THIS PROGRAM


                    WHICH: 2
 NAME                        PHONE NUMBER
----------------------------------------
A.A.........................002 771 0000
B.B.........................032 324 5678
THAT'S ALL OF THEM! - PRESS 'RETURN'
----------------------------------------
----------------------------------------
```

若要修改或消去其中某些名單，則可選擇4或5，此時由螢幕要求使用者將欲更改或消去的名字輸入，如果只把字首輸入螢幕卽將其資料列出並詢問是否正確、是否確定修改或消去，如果不是所要修改或消去的名單（可能按鍵疏忽等）則回答“N”，螢幕繼續把同字首的下一個名字列出。

```
PROGRAM CURRENTLY CONTAINS 6 LISTINGS
ROOM FOR 144 ADDITIONAL LISTINGS
THE PRINTER IS 'ON'
YOU MAY CHOOSE FROM THE FOLLOWING:


1  SEARCH FOR A LISTING
2  LIST THE ENTIRE FILE
3  ENTER NEW LISTINGS
4  DELETE A LISTING
5  EDIT A LISTING
6  PRINTER ON/OFF
7  EXIT THIS PROGRAM


                    WHICH: 4
PLEASE ENTER THE NAME OF THE PARTY YOU
WISH DELETED FROM THE DIRECTORY AND
PRESS 'RETURN'.
-> A
APPLE COMPUTER INC...........408 996 1010
DO YOU WISH TO DELETE THIS DATA (Y/N) N
A............................234 567 8900
DO YOU WISH TO DELETE THIS DATA (Y/N) Y
```

如果要求印字機將結果印出，必須先選第六個項目：

```
PROGRAM CURRENTLY CONTAINS 4 LISTINGS
ROOM FOR 146 ADDITIONAL LISTINGS
THE PRINTER IS  OFF
YOU MAY CHOOSE FROM THE FOLLOWING:


1  SEARCH FOR A LISTING
2  LIST THE ENTIRE FILE
3  ENTER NEW LISTINGS
4  DELETE A LISTING
5  EDIT A LISTING
6  PRINTER ON/OFF
7  EXIT THIS PROGRAM


                    WHICH: 6
THE PRINTER IS 'ON'
```

如果程式已使用完畢，可選擇第七個項目來停止程式，如果在執行程式過程中曾經更改過名單的內容，則電腦會要求使用者將最後的結果存錄到錄

音帶或磁碟上，如果並未更改內容，則程式自動結束。

```
------------------------------------------------------------
PROGRAM CURRENTLY CONTAINS 5 LISTINGS
ROOM FOR 145 ADDITIONAL LISTINGS
THE PRINTER IS 'ON'
YOU MAY CHOOSE FROM THE FOLLOWING:


1  SEARCH FOR A LISTING
2  LIST THE ENTIRE FILE
3  ENTER NEW LISTINGS
4  DELETE A LISTING
5  EDIT A LISTING
6  PRINTER ON/OFF
7  EXIT THIS PROGRAM


                    WHICH: 7
INSERT A NEW TAPE INTO THE RECORDER, SETON RECORD,
AND PRESS 'RETURN'
THE PROGRAM AND DATA HAVE BEEN SAVED!!

]
```

程式十二 吃角子老虎

利用電腦可以模仿各種實物的動作與過程，不管是理論的計算或實物的模擬，譬如是飛機的起落、行星運行、都市交通的模擬甚或本文所提到的玩樂遊戲——吃角子老虎機器。和上一個程式“賽馬”的例子類似的，這都是屬於遊戲程式，因此其結構相仿，藉此我們可找其相似的規則，或許可加以利用發展出新的程式吧！

程 式

```
]LIST

10  REM  MICROCOMPUTER SOFTWARE ASS
     OCIATES
15  REM  HAYWARD, CALIFORNIA 94541
20  REM  MSA CASINO/SLOTS 781019PLQ

25  REM  COPYRIGHT (C) 1978 BY MSA
30  HOME : GOSUB 2010: HOME
32  PRINT "        THE PAYOUTS ARE AS
      FOLLOWS:": PRINT
35  PRINT "     G/2      G/2      G/2
         PAYS  $50"
37  PRINT "    BELL     BELL     BELL
         PAYS  $18"
```

```
38  PRINT "   BELL    BELL     G/2
        PAYS  $18"
39  PRINT "   PLUM    PLUM    PLUM
        PAYS  $14"
40  PRINT "   PLUM    PLUM     G/2
        PAYS  $14"
41  PRINT "  ORANGE  ORANGE  ORANGE
        PAYS  $10"
42  PRINT "  ORANGE  ORANGE    G/2
        PAYS  $10"
43  PRINT "  CHERRY  CHERRY  ******
        PAYS  $5"
44  PRINT "  CHERRY  ******  ******
        PAYS  $2": PRINT
45  INVERSE : PRINT  TAB( 40): NORMAL

50 T =  - 1:X = TT
80  PRINT "  ";: INVERSE : PRINT "R
     EEL";: POKE 36,10: PRINT "REEL
     ";: POKE 36,18: PRINT "REEL";:
      POKE 36,25: PRINT "PLAY";: POKE
     36,33: PRINT "WIN": NORMAL
81  PRINT " ";: INVERSE : PRINT "
     1 ";: POKE 36,10: PRINT "  2 "
     ;: POKE 36,18: PRINT "  3 ";: POKE
     36,25: PRINT "PLAY";: POKE 36,
     33: PRINT "WIN": NORMAL
82  VTAB 19: INVERSE : PRINT  TAB(
     40): NORMAL
85  VTAB 17
200 Q =  INT ( RND (1.13) * 15):R =
      INT ( RND (2.12) * 15):S =  INT
```

```
      ( RND (3.13) * 15): GOTO 600
400   HOME
490   REM ARK: REEL  VALUES
500   PRINT "   G/2   ";: RETURN
502   PRINT "  BELL   ";: RETURN
504   PRINT "  PLUM   ";: RETURN
506   PRINT " ORANGE ";: RETURN
508   PRINT " CHERRY ";: RETURN
510   FOR DD = 1 TO 750: NEXT : RETURN

588 REMARK: Q = REEL 1
590   GOSUB 510
600   IF  Q = 0  THEN  GOSUB  500
602   IF Q =  > 1 AND Q <  = 2 THEN
      Q = 1: GOSUB 502
604   IF Q =  > 3 AND Q <  = 5 THEN
      Q = 2: GOSUB 504
606   IF Q =  > 6 AND Q <  = 9 THEN
      Q = 3: GOSUB 506
608   IF Q =  > 10 AND Q <  = 15 THEN
      Q = 4: GOSUB 508
688 REM R = REEL 2
690   GOSUB 510
700   IF  R = 0  THEN  GOSUB  500
702   IF R =  > 1 AND R <  = 2 THEN
      R = 1: GOSUB 502
704   IF R =  > 3 AND R <  = 5 THEN
      R = 2: GOSUB 504
706   IF R =  > 6 AND R <  = 9 THEN
      R = 3: GOSUB 506
708   IF R =  > 10 AND R <  = 15 THEN
      R = 4: GOSUB 508
```

```
788  REM  S=REEL3
790  GOSUB 510
800  IF  S = 0  THEN  GOSUB  500
802  IF S =  > 1 AND S <  = 2 THEN
     S = 1: GOSUB 502
804  IF S =  > 3 AND S <  = 5 THEN
     S = 2: GOSUB 504
806  IF S =  > 6 AND S <  = 9 THEN
     S = 3: GOSUB 506
808  IF S =  > 10 AND S <  = 15 THEN
     S = 4: GOSUB 508
810  GOSUB 510
900 U = 0
910  IF Q = 0 AND R = 0 AND S = 0 THEN
     U = 50
920  IF Q = 1 AND R = 1 AND S = 1 THEN
     U = 18
930  IF Q = 1 AND R = 1 AND S = 0 THEN
     U = 18
940  IF Q = 2 AND R = 2 AND S = 2 THEN
     U = 14
950  IF Q = 2 AND R = 2 AND S = 0 THEN
     U = 14
960  IF Q = 3 AND R = 3 AND S = 3 THEN
     U = 10
970  IF Q = 3 AND R = 3 AND S = 0 THEN
     U = 10
980  IF Q = 4 THEN U = 2
990  IF Q = 4 AND R = 4 THEN U = 5
1450 X = T + U + X
1460  PRINT "   ";T;"      ";U
1461  VTAB 21: PRINT " TOTAL CASH $
```

```
       ";X
1475  IF  X = 0  THEN  4000
1478  PRINT " AGAIN, ";N$;
1480 A =  PEEK ( - 16384): IF A < 1
     27 THEN 1480
1481  INPUT A$: POKE  - 16368,0
1482  IF A$ = "N" THEN 4000
1485  IF A$ = "Y" THEN 1486
1486  VTAB 21: CALL  - 958: VTAB 17
     : PRINT  TAB( 40): VTAB 17: GOTO
     200
2000  HOME : PRINT : PRINT : PRINT
     : PRINT
2010  PRINT  TAB( 12)"G O O D  D A
     Y ! !": PRINT : PRINT
2020  PRINT  TAB( 5)"WELCOME TO THE
      SLOT MACHINE AT": PRINT  TAB(
     12)"THE G/2 CASINO!": PRINT : PRINT

2030  INPUT "WHAT'S YOUR NAME? ";N$

2035  PRINT : PRINT "I'M A DOLLAR M
     ACHINE"
2040  INPUT "HOW MANY DO YOU HAVE T
     O FEED ME? ";TT:TT =  INT (TT)

2050  IF TT <  = 0 THEN TT = 10
2060  PRINT : PRINT "TO 'PULL' THE
     HANDLE, TYPE 'Y'": PRINT "TO Q
     UIT, TYPE 'N'."
2070  FOR DD = 1 TO 3000: NEXT
3000  RETURN
```

```
4000  HOME
4010  PRINT "    WELL, ";N$",": PRINT

4030  PRINT "    YOU STARTED WITH $"
      ;TT;"."
4040  PRINT "    YOU NOW HAVE $";X;"
      .": PRINT
4060  PRINT "    WE HOPE YOU LIKED Y
      OUR 'PULL' AT OUR"
4070  PRINT "    SLOT MACHINE. COME
      BACK AGAIN SOON!"
4080  PRINT
4090  PRINT "    PERHAPS YOU COULD T
      RY YOUR LUCK AT"
4100  PRINT "    ONE OF OUR OTHER TA
      BLES.  BUT, BE"
4110  PRINT "    CAREFUL OF THE CRAP
      S TABLE. IT GETS"
4120  PRINT "    A LITTLE ROWDY!": PRINT
      : PRINT
9999  END
```

程式説明

當程式執行開始，由 2010 行至 3000 行的副程式來與使用者交談，並設定參加遊戲的金額或點數（ INPUT TT ）。第 32 行至 45 行印出得獎的格式與獎金倍數。第 80 行至第 85 行設定畫面格式。

由第200行以 RND 函數來任意造出Q、R、S三個其值介於 0 至15之間的整數亂數。第600至608行、第700至708行以及800至第 808行用來判斷Q、R、S的值，如果其值爲 0 則由500行印出“ G/2 ”字樣，如果介於 0 與 2 之間則設其值爲 1 並由第502行印出“ BELL ”字樣，若

介於 3 與 5 之間則設其值爲 2 並由第 504 行印出“ PLUM ”字樣，若介於 6 與 9 之間則設其值爲 3 並由第 506 行印出“ ORANGE ”字樣，其餘則設其值爲 4 並由第 508 行印出“ CHERRY ”字樣。

當各亂數Q、R、S之值分別設爲0、1、2、3、4之間的一個值，並且在畫面顯示各種字樣之後，由第900行至990行的得獎表格來對照出是否得獎，若是則設定U的值爲獎金，若R、Q、S不在得獎表格內，則U的值爲0，至此由第1450行求出參加遊戲者所餘金額或點數，其中T＝－1是每次遊戲必須使用掉一點，最終點數是上次剩餘的點數扣掉使用掉的一點再加上得獎的點數即X＝T＋U＋X。

當最終X點數算出後，即印出至螢幕，如果X＝0則程式轉至第4000行而結束，若X≠0則由1478行至1486行來詢問使用者繼續與否。

程式改進

本程式具備了吃角子老虎（SLOT MACHINE）的結構，其運作工能上說來，是與實體無異，只是在顯示上的功能不足，若能利用APPLE的畫面能力來取代字幕上的顯示，則更能達到視覺上的效果。

要做如此的改良，可參照本書有關高解像畫面的結構來安排出各種圖案，然後由Q、R、S的值來判斷需要顯示何種畫面。總之，只要用各種不同的畫面來取代由第500行至508行的PRINT指令就行了，當然整個畫面的印出格式也是要重新安排的。

程式十三　APPLE 輪盤

上一個程式的吃角子老虎（SLOT　MACHINE），是比較易於模擬的遊戲，如果是像輪盤這種東西就比較複雜了。本例子就是模擬一個輪盤操作的情形，其目的是讓讀者更進一步的明白複雜情況的模擬，爲了保持大家閱讀的興趣，在此我們就不做繁長的解說了，只要仿上一個例子的方式，即可掌握程式工作的情形，因此下面就只列出程式本身以供參考。

程　式

```
90  HOME : PRINT : PRINT : PRINT : PRINT

100  PRINT  TAB( 10);"G/2  R O U L
     E T T E"
110  DIM CT(30,22),M$(18),NM$(38),W
     $(38)
120  FOR I = 1 TO 18: READ M$(I): NEXT

130  FOR I = 1 TO 38: READ NM$(I): NEXT

140  FOR I = 1 TO 38: READ W$(I): NEXT

150  DATA "STRAIGHT-UP","SPLIT","3-
     NUMBER","CORNER","5-NUMBER"
160  DATA "6-NUMBER","1ST DOZEN","2
```

```
     ND DOZEN","3RD DOZEN"
170  DATA "1ST COLUMN","2ND COLUMN"
     ,"3RD COLUMN","RED","BLACK"
180  DATA "EVEN","ODD","HIGH","LOW"
190  DATA G,R,B,R,B,R,B,R,B,R,B,B,R
     ,B,R,B,R,B,R,R,B,R
200  DATA B,R,B,R,B,R,B,B,R,B,R,B,R
     ,B,R,G
210  DATA 0G,2B,14R,35B,23R,4B,16R,
     33B,21R,6B,18R,31B,19R
220  DATA 8B,12R,29B,25R,10B,27R,00
     G,1R,13B,36R,24B,3R,15B
230  DATA 34R,22B,5R,17B,32R,20B,7R
     ,11B,30R,26B,9R,28B
240  PRINT : INPUT "WHAT IS YOUR NA
     ME? ";N$
250  PRINT : INPUT "WHAT IS YOUR ST
     ARTING STAKE? ";T
260  IF T <  = 0 THEN  POKE 37, PEEK
     (37) - 2: GOTO 250
270  PRINT : PRINT N$;", DO YOU WIS
     H TO SEE": INPUT "THE INSTRUCT
     IONS? ";Y$
280  IF Y$ = "N" THEN 800
290  IF Y$ <  > "Y" THEN  POKE 37, PEEK
     (37) - 3: GOTO 270
300  HOME : PRINT "WE WILL BE PLAYI
     NG WITH THE AMERICAN"
310  PRINT "WHEEL WHICH HAS 38 POCK
     ETS: ONE MARKED"
320  PRINT "ZERO, ONE MARKED DOUBLE
      ZERO AND THE"
```

```
330  PRINT "REST MARKED 1 THROUGH 3
     6. THE 0 AND 00"
340  PRINT "ARE GREEN WHILE THE RES
     T ARE DIVIDED"
350  PRINT "EQUALLY BETWEEN RED AND
      BLACK.  ON THE"
360  PRINT "DISPLAY, RED NUMBERS AR
     E NORMAL AND"
370  PRINT "BLACK ARE REVERSED.": PRINT

380  PRINT "ALL BETS WILL BE IN DOL
     LAR INCREMENTS"
390  PRINT "AND CAN BE PLACED ON AN
     Y OR ALL OF THE"
400  PRINT "FOLLOWING TYPES OF BETS
     :": PRINT
410  PRINT "   STRAIGHT UP        35
     TO 1"
420  PRINT "      A BET THAT THIS 1
     NUMBER WILL BE"
430  PRINT "      ROLLED.": PRINT
440  PRINT "   SPLIT BET        17 TO
      1"
450  PRINT "      A BET THAT 2 ADJAC
     ENT NUMBERS"
460  PRINT "      EITHER HORIZONTALL
     Y OR VERTICALLY"
470  PRINT "      WILL BE ROLLED. EG
      10+13 BUT NOT"
480  PRINT "      10+14": GOSUB 780
490  PRINT : PRINT : PRINT "   3-NU
     MBER BET       11 TO 1"
```

```
500  PRINT "      A BET THAT ONE OF
     3 CONSECUTIVE"
510  PRINT "      NUMBERS WILL BE RO
     LLED. EG 7+8+9"
520  PRINT "      OR 0+1+2 BUT NOT 5
     +6+7."
525  PRINT : PRINT "   CORNER BET
         8 TO 1"
530  PRINT "      A BET THAT ONE OF
     4 NUMBERS OF"
535  PRINT "      A SQUARE WILL BE R
     OLLED. EG"
540  PRINT "      1+2+4+5 OR 26+27+2
     9+30, BUT NOT"
545  PRINT "      13+14+15+16."
550  PRINT : PRINT "   5-NUMBER BET
           6 TO 1"
555  PRINT "      THE ONLY  5-NUMBER
      BET IS"
556  PRINT "      0+00+1+2+3."
560  PRINT : PRINT "   6-NUMBER BET
           5 TO 1"
565  PRINT "      A BET THAT ONE OF
     2 CONSECUTIVE"
570  PRINT "      ROWS WILL BE ROLLE
     D. EG 1-6 OR"
575  PRINT "      19-24, BUT NOT 2-7
     .": GOSUB 780
580  PRINT : PRINT : PRINT "   1ST,
      2ND OR 3RD DOZEN   2 TO 1"
585  PRINT "      A BET THAT THE NUM
     BER ROLLED WILL"
```

```
590  PRINT "      BE IN ONE OF THES
     E GROUPS: 1-12,"
595  PRINT "     13-24 OR 25-36."
600  PRINT : PRINT "   1ST, 2ND OR
     3RD COLUMN    2 TO 1"
605  PRINT "     A BET THAT THE NUM
     BER WILL BE IN"
610  PRINT "     1 OF THESE COLUMNS
     ."
620  PRINT : PRINT "   RED/BLACK
       1 TO 1"
630  PRINT : PRINT "   EVEN/ODD
       1 TO 1"
640  PRINT : PRINT "   HIGH/LOW
       1 TO 1"
650  PRINT "     A BET THAT THE NUM
     BER ROLLED WILL"
660  PRINT "     BE LOW (1-18) OR H
     IGH (19-36).": GOSUB 780
760  PRINT : PRINT : PRINT : PRINT
     "GOOD LUCK, ";N$;". YOU'LL NEE
     D IT!!"
770  GOSUB 3280: GOTO 800
780  PRINT : PRINT "HIT 'RETURN' WH
     EN READY";: INPUT Y$
790  HOME : RETURN
800  REM  PRINT TABLE LAYOUT
810  IF BN = 0 THEN BN = 1
820  HOME : PRINT : PRINT : PRINT "
     ------------------------------
     ---------"
830  PRINT "     3 ";: INVERSE : PRINT
```

```
     " 6";: NORMAL : PRINT "  9 12
     ";: INVERSE : PRINT "15";: NORMAL

832  PRINT " 18 21 ";: INVERSE : PRINT
     "24";: NORMAL : PRINT " 27 30
     ";: INVERSE : PRINT "33";: NORMAL
     : PRINT " 36"
840  PRINT " OO ---------------------
     -----------------"
850  PRINT "     ";: INVERSE : PRINT
     " 2";: NORMAL : PRINT "  5 ";:
      INVERSE : PRINT " 8";: NORMAL
     : PRINT " ";: INVERSE : PRINT
     "11";: NORMAL
852  PRINT " 14 ";: INVERSE : PRINT
     "17";: NORMAL : PRINT " ";: INVERSE
     : PRINT "20";: NORMAL : PRINT
     " 23 ";: INVERSE : PRINT "26";

854  NORMAL : PRINT " ";: INVERSE :
      PRINT "29";: NORMAL : PRINT "
      32 ";: INVERSE : PRINT "35": NORMAL

860  PRINT "  O ---------------------
     -----------------"
870  PRINT "      1 ";: INVERSE : PRINT
     " 4";: NORMAL : PRINT "  7 ";:
      INVERSE : PRINT "10";: NORMAL
     : PRINT " ";
872  INVERSE : PRINT "13";: NORMAL
     : PRINT " 16 19 ";: INVERSE : PRINT
     "22";: NORMAL : PRINT " 25 ";:
```

```
INVERSE
874  PRINT "28";: NORMAL : PRINT "
     ";: INVERSE : PRINT "31";: NORMAL
     : PRINT " 34"
890  PRINT "--------------------------
     ------------------"
900  PRINT : PRINT " 1. STRAIGHT UP
     " TAB( 21)"2. SPLIT"
910  PRINT " 3. 3-NUMBER" TAB( 21)"
     4. CORNER"
920  PRINT " 5. 5-NUMBER" TAB( 21)"
     6. 6-NUMBER"
930  PRINT " 7. 1,2,3 DOZEN" TAB( 2
     1)"8. 1,2,3 COLUMN"
940  PRINT " 9. RED/BLACK" TAB( 20)
     "10. EVEN/ODD"
950  PRINT "11. HIGH/LOW" TAB( 20)"
     12. LIST BETS"
960  PRINT "13. ROLL" TAB( 20)"14.
     UP STAKES"
970  PRINT "15. QUIT" TAB( 20)"16.
     INSTRUCTIONS"
975  PRINT : PRINT
980  GOSUB 3270
990  INPUT "PLACE YOUR BET: ";C
1000  ON C GOTO 1040,1110,1290,1460
     ,1580,1640,1760,1850,1930
1010 C = C - 9
1020  ON C GOTO 2030,2140,2240,2550
     ,2450,2500,1025
1025  HOME : GOTO 410
1030  GOTO 3230
```

```
1040  REM  STRAIGHT-UP BET
1050  GOSUB 3270: PRINT M$(1); TAB(
     20);"#=";: GOSUB 3460
1060  IF NM < 0 THEN 3230
1070  GOSUB 3290: IF B = 0 THEN 990

1080  IF NM < 0 OR NM > 37 THEN 323
     0
1090 CT(BN,1) = 1:CT(BN,3) = 35:CT(
     BN,4) = 1
1100 CT(BN,5) = NM: GOTO 3400
1110  REM   SPLIT BET
1120  GOSUB 3270: PRINT M$(2); TAB(
     18);"1ST# =";: GOSUB 3460
1130 N1 = NM: IF N1 < 0 THEN 3230
1140 X = 27: GOSUB 3270: PRINT "2ND
     # =";: GOSUB 3460
1150 N2 = NM: IF N2 < 0 THEN 3230
1160  GOSUB 3290: IF B = 0 THEN 990

1170  IF N1 < 0 OR N1 > 37 THEN 323
     0
1180  IF N2 < 0 OR N2 > 37 THEN 323
     0
1190  IF N1 = N2 THEN 3230
1200  GOSUB 3430
1210  IF N2 - N1 = 3 THEN 1270
1220  IF N2 - N1 = 1 AND N1 - ( INT
     (N1 / 3) * 3) > 0 THEN 1270
1230  IF N2 = 37 AND N1 = 0 THEN 12
     70
1240  IF N2 = 1 AND N1 = 0 THEN 127
```

```
      0
1250  IF N2 = 2 AND N1 = 0 THEN 127
      0
1260  GOTO 3230
1270 CT(BN,1) = 2:CT(BN,3) = 17:CT(
      BN,4) = 2
1280 CT(BN,5) = N1:CT(BN,6) = N2: GOTO
      3400
1290  REM   3-NUMBER BET
1300  GOSUB 3270: PRINT M$(3); TAB(
      18);"1ST# =";: GOSUB 3460
1310 N1 = NM: IF N1 < 0 THEN 3230
1320 X1 = 22
1330 X = X1 +  LEN (X$): GOSUB 3270
      : PRINT "3RD# =";: GOSUB 3460
1340 N2 = NM: IF N2 < 0 THEN 3230
1350  GOSUB 3290: IF B = 0 THEN 990

1360  GOSUB 3430: IF N1 = N2 THEN 3
      230
1370  IF N2 - ( INT (N2 / 3) * 3) =
      0 AND N2 - N1 = 2 THEN 1420
1380  IF N2 = 2 AND N1 = 0 THEN N3 =
      1: GOTO 1430
1390  IF N2 = 37 AND N1 = 2 THEN N3
       = 0: GOTO 1430
1400  IF N2 = 37 AND N1 = 3 THEN N3
       = 2: GOTO 1430
1410  GOTO 3230
1420 N3 = N2:N2 = N1 + 1
1430 CT(BN,5) = N1:CT(BN,6) = N2:CT
      (BN,7) = N3
```

```
1440 CT(BN,1) = 3:CT(BN,3) = 11:CT(
     BN,4) = 3
1450  GOTO 3400
1460  REM   CORNER BETS
1470  GOSUB 3270: PRINT M$(4); TAB(
     18);"1ST# =";: GOSUB 3460
1480 N1 = NM: IF N1 < 0 THEN 3230
1490 X = 22 +  LEN (X$): GOSUB 3270
     : PRINT "4TH# =";: GOSUB 3460
1500 N2 = NM: IF N2 < 0 THEN 3230
1510  GOSUB 3290: IF B = 0 THEN 990

1520  GOSUB 3430: IF N1 = N2 THEN 3
     230
1530  IF N2 - ( INT (N2 / 3) * 3) <
      > 1 AND N2 - N1 = 4 THEN 1550

1540  GOTO 3230
1550 CT(BN,1) = 4:CT(BN,3) = 8:CT(B
     N,4) = 4
1560 CT(BN,5) = N1:CT(BN,6) = N1 +
     1:CT(BN,7) = N2 - 1
1570 CT(BN,8) = N2: GOTO 3400
1580  REM   5-NUMBER BET
1590  GOSUB 3270: PRINT M$(5); TAB(
     18);"#=0,00,1,2 OR 3": GOSUB 3
     280
1600  GOSUB 3290: IF B = 0 THEN 970

1610 CT(BN,1) = 5:CT(BN,3) = 6:CT(B
     N,4) = 5:CT(BN,5) = 0
1620 CT(BN,6) = 1:CT(BN,7) = 2:CT(P
```

```
     N,8) = 3:CT(BN,9) = 37
1630  GOTO 3400
1640  REM   6-NUMBER BETS
1650  GOSUB 3270: PRINT M$(6); TAB(
     18);"1ST# =";: GOSUB 3460
1660 N1 = NM: IF N1 < 0 THEN 3230
1670 X = 22 +  LEN (X$): GOSUB 3270
     : PRINT "6TH# =";: GOSUB 3460
1680 N2 = NM: IF N2 < 0 THEN 3230
1690  GOSUB 3290: IF B = 0 THEN 990

1700  GOSUB 3430: IF N1 = N2 THEN 3
     230
1710  IF N2 - ( INT (N2 / 3) * 3) =
     0 AND N2 - N1 = 5 THEN 1730
1720  GOTO 3230
1730 CT(BN,1) = 6:CT(BN,3) = 5:CT(B
     N,4) = 6
1740  FOR I = 1 TO 6:CT(BN,4 + I) =
     N1 + I - 1: NEXT
1750  GOTO 3400
1760  REM   1ST,2ND,3RD DOZEN
1770  GOSUB 3270: PRINT "1,2,3 DOZE
     N"; TAB( 19);: INPUT "WHICH DO
     ZEN? ";NM
1780  IF NM = 1 OR NM = 2 OR NM = 3
      THEN 1800
1790  GOTO 3230
1800  GOSUB 3290: IF B = 0 THEN 990

1810 CT(BN,1) = 6 + NM:CT(BN,3) = 2
     :CT(BN,4) = 12
```

```
1820 J = NM * 12 - 11
1830  FOR I = 1 TO 12:CT(BN,4 + I) =
     J - 1 + I: NEXT
1840  GOTO 3400
1850  REM   1ST,2ND,3RD COLUMN
1860  GOSUB 3270: PRINT "1,2,3 COLU
     MN"; TAB( 19);: INPUT "WHICH C
     OLUMN? ";NM
1870  IF NM = 1 OR NM = 2 OR NM = 3
      THEN 1890
1880  GOTO 3230
1890  GOSUB 3290: IF B = 0 THEN 990

1900 CT(BN,1) = 9 + NM:CT(BN,3) = 2
     :CT(BN,4) = 12
1910  FOR I = 1 TO 12:CT(BN,4 + I) =
     NM:NM = NM + 3: NEXT
1920  GOTO 3400
1930  REM   RED/BLACK
1940  GOSUB 3270: PRINT "RED/BLACK"
     ; TAB( 19);: INPUT "WHICH COLO
     R? ";Y$
1950  IF Y$ = "R" THEN NM = 13: GOTO
     1980
1960  IF Y$ <  > "B" THEN 3230
1970 NM = 14
1980  GOSUB 3290: IF B = 0 THEN 990

1990 CT(BN,1) = NM:CT(BN,3) = 1:CT(
     BN,4) = 18:J = 5
2000  FOR I = 1 TO 37: IF NM$(I) <
      > Y$ THEN 2020
```

```
2010 CT(BN,J) = I - 1:J = J + 1
2020  NEXT : GOTO 3400
2030  REM   EVEN/ODD
2040  GOSUB 3270: PRINT "EVEN/ODD";
      TAB( 19);: INPUT "WHICH ONE?
     ";Y$
2050  IF Y$ = "E" THEN NM = 15: GOTO
     2080
2060  IF Y$ <  > "O" THEN 3230
2070 NM = 16
2080  GOSUB 3290: IF B = 0 THEN 990

2090 CT(BN,1) = NM:CT(BN,3) = 1:CT(
     BN,4) = 18:J = 5
2100 NM = NM - 15
2110  FOR I = 1 TO 36: IF I - ( INT
     (I / 2) * 2) <  > NM THEN 2130

2120 CT(BN,J) = I:J = J + 1
2130  NEXT : GOTO 3400
2140  REM   HI/LOW
2150  GOSUB 3270: PRINT "HIGH/LOW";
      TAB( 19);: INPUT "WHICH ONE?
     ";Y$
2160  IF Y$ = "H" THEN NM = 17: GOTO
     2190
2170  IF Y$ <  > "L" THEN 3230
2180 NM = 18
2190  GOSUB 3290: IF B = 0 THEN 990

2200 CT(BN,1) = NM:CT(BN,3) = 1:CT(
     BN,4) = 18:J = 4
```

```
2210 NM = (18 - NM) * 18
2220  FOR I = 1 TO 18:CT(BN,J + I) =
     NM + I: NEXT
2230  GOTO 3400
2240  REM   LIST ALL BETS
2250 X1 = 1
2260 X2 = 13
2270 X3 = 22
2280 X4 = 31
2290 TB = 0
2300  HOME : PRINT "NAME"; TAB( X2)
     ;"#'S";
2310  PRINT  TAB( X3);"$ BET"; TAB(
     X4);"ODDS"
2320 J = 40
2330  FOR I = 1 TO J: PRINT "-";: NEXT

2340  PRINT
2350  IF BN = 1 THEN  PRINT "NO BET
     S MADE": GOTO 2420
2360  FOR I = 1 TO BN - 1: PRINT M$
     (CT(I,1));
2370  PRINT  TAB( X2);:Q = CT(I,5):
      GOSUB 3520
2380  IF CT(I,4) = 1 THEN 2400
2390  PRINT "-";:Q = CT(I,4 + CT(I,
     4)): GOSUB 3520
2400  PRINT  TAB( X3);CT(I,2); TAB(
     X4);CT(I,3);"-1"
2410 TB = TB + CT(I,2): NEXT
2420  PRINT : PRINT "CURRENT STAKE=
     $"T: PRINT "DOLLARS BET=$";TB
```

```
2430  PRINT : INPUT "HIT 'RETURN' W
     HEN READY";Y$
2440  GOTO 820
2450  REM   UP THE STAKES
2460  GOSUB 3270: INPUT "HOW MUCH I
     S YOUR NEW STAKE? ";T
2470  GOSUB 3270: IF T > 0 THEN 990

2480  PRINT "YOU MUST HAVE SOME MON
     EY ON THE TABLE!"
2490  GOSUB 3280: GOSUB 3270: PRINT
     : GOTO 2460
2500  REM   QUIT GAME
2510  HOME : PRINT : PRINT
2520  PRINT "THANK YOU FOR PLAYING
     ROULETTE, ";N$"."
2530  PRINT "SEE YOU LATER."
2540  END
2550  REM   DRAW & SPIN THE WHEEL
2560  HOME : PRINT : PRINT : PRINT
     : PRINT : PRINT
2580  FOR I = 1 TO 10: GOSUB 2666
2590  PRINT  TAB( 3 * (I - 1) + 5);
     P$;: NORMAL : PRINT " ";
2600  NEXT : PRINT
2610  FOR K = 11 TO 19:I = 49 - K
2620  PRINT "      ";: GOSUB 2666: PRINT
     P$;: NORMAL : PRINT  TAB( 32);

2630 I = K: GOSUB 2666: PRINT P$: NORMAL

2640  NEXT
```

```
2650  FOR K = 1 TO 10:I = 30 - K: GOSUB
     2666
2660  PRINT  TAB( 3 * (K - 1) + 5);
     P$;: NORMAL : PRINT " ";
2661  NEXT : PRINT : GOTO 2680
2666 L1 =  LEN (W$(I)) - 1:P$ =  LEFT$
     (W$(I),L1): IF L1 = 1 THEN P$ =
     " " + P$
2667  IF  RIGHT$ (W$(I),1) = "B" THEN
      INVERSE
2668  RETURN
2680 SS =  - 49:NN =  INT ( RND (1)
      * 38 + 1)
2685 V = 33:H = 4
2690 V = V - 2: GOSUB 2800: FOR I =
     1 TO 10:SS = SS + 1: IF SS = N
     N THEN 2830
2700 V = V - 2: GOSUB 2800: NEXT :V
      = V - 2: GOSUB 2800
2710  FOR I = 1 TO 10:SS = SS + 1: IF
     SS = NN THEN 2830
2720 H = H + 3: GOSUB 2800: NEXT
2730 V = V + 4: GOSUB 2800: FOR I =
     1 TO 10:SS = SS + 1: IF SS = N
     N THEN 2830
2740 V = V + 2: GOSUB 2800: NEXT
2750  FOR I = 1 TO 10:SS = SS + 1: IF
     SS = NN THEN 2830
2760 H = H - 3: GOSUB 2800: NEXT
2770  GOTO 2690
2781 H = H + 1: COLOR= 2: PLOT H,V:
      COLOR= 10: PLOT H,V: RETURN
```

```
2785  REM  LEFT
2786 H = H - 1: COLOR= 2: PLOT H,V:
      COLOR= 10: PLOT H,V: RETURN
2790  REM  UP
2791 V = V - 2: COLOR= 2: PLOT H,V:
      COLOR= 10: PLOT H,V: RETURN
2795  REM  DOWN
2796 V = V + 2: COLOR= 2: PLOT H,V:
      COLOR= 10: PLOT H,V: RETURN
2800  COLOR= 2: PLOT H,V: FOR X = 1
      TO 50: NEXT : COLOR= 10: PLOT
     H,V: RETURN
2820  REM   DO PAYOFFS
2830 Q$ =  LEFT$ (W$(NN), LEN (W$(N
     N)) - 1)
2840  IF  RIGHT$ (W$(NN),1) = "G" THEN
     Q1$ = "GREEN"
2850  IF  RIGHT$ (W$(NN),1) = "R" THEN
     Q1$ = "RED"
2860  IF  RIGHT$ (W$(NN),1) = "B" THEN
     Q1$ = "BLACK"
2870  POKE 37,9: PRINT : POKE 36,16

2890  PRINT Q$;" ";Q1$;
2900  GOSUB 3280: HOME
2910  PRINT : PRINT Q$;" ";Q1$
2920 X1 = 1
2930 X2 = 17
2940 X3 = 23
2950 X4 = 29
2960 X5 = 40
2970  PRINT
```

```
2980  PRINT  TAB( X1);"NAME"; TAB(
      X2 - 1);"$ BET"; TAB( X3);"ODD
      S";
2990  PRINT  TAB( X4);"PAYOFFS"
3000  IF BN = 1 THEN  PRINT : PRINT
      "NO BETS MADE": GOTO 3210
3010  FOR I = 1 TO BN - 1
3020  PRINT M$(CT(I,1));"(";:Q = CT
      (I,5): GOSUB 3520
3030  IF CT(I,4) > 1 THEN 3050
3040  PRINT ")";: GOTO 3060
3050  PRINT "-";:Q = CT(I,4 + CT(I,
      4)): GOSUB 3520: PRINT ")";
3060  PRINT  TAB( X2);CT(I,2); TAB(
      X3);CT(I,3);"-1";
3070  IF Q$ = "00" THEN N = 37: GOTO
      3100
3080 N =  VAL (Q$)
3090  PRINT  TAB( X4);
3100 J = 5:J1 = CT(I,4)
3110  IF CT(I,J) = N THEN 3150
3120 J = J + 1:J1 = J1 - 1
3130  IF J1 > 0 THEN 3110
3140  PRINT "LOSES $";:ST = CT(I,2)
      : GOTO 3170
3150  PRINT "WINS $";:ST = CT(I,2) +
      CT(I,2) * CT(I,3)
3160 T = T + ST
3170  PRINT ST
3180  NEXT
3190 X6 = 27
3200  PRINT : PRINT "REMAINING STAK
```

```
     E IS $";T
3210  PRINT : INPUT "HIT 'RETURN' W
     HEN READY";Y$
3220 BN = 1: GOTO 800
3230  REM   ILLEGAL #
3240 X = 0:Y = 14: GOSUB 3270
3250  FLASH : PRINT "ILLEGAL COMMAN
     D!": NORMAL : GOSUB 3280
3260  GOSUB 3270: GOTO 990
3270  POKE 37,19: CALL  - 958
3280  FOR I = 1 TO 1500: NEXT : RETURN

3290  REM  CHECK BET
3300 X = 40
3310  GOSUB 3270: INPUT "BET $";B
3320  IF  INT (B) = B THEN 3350
3330 X = 0:Y = 14: GOSUB 3270
3340  PRINT "DOLLAR BETS ONLY, PLEA
     SE!!": GOTO 3380
3350  IF B <  = T THEN  RETURN
3360 X = 0:Y = 14: GOSUB 3270
3370  PRINT "INSUFFICIENT MONEY TO
     PLACE THIS BET!"
3380  GOSUB 3280: GOSUB 3270
3390 B = 0: RETURN
3400  REM   NEXT
3410 CT(BN,2) = B:BN = BN + 1
3420 T = T - B: GOTO 980
3430  REM   SWITCH
3440  IF N1 < N2 THEN  RETURN
3450 N3 = N2:N2 = N1:N1 = N3: RETURN
```

```
3460 REM   INPUT CHAR & CONVERT
3470  INPUT " ";X$: IF X$ = "OO" THEN
     NM = 37: RETURN
3480  IF X$ = "O" THEN NM = O: RETURN

3490 NM =  VAL (X$)
3500  IF NM > O AND NM < 37 THEN  RETURN

3510 NM =  - 1: RETURN
3520  IF Q = 37 THEN  PRINT " OO ";
     : RETURN
3530  PRINT Q;: RETURN
3540  PRINT "ILLEGAL COMMAND!";: GOSUB
     3280
```

程式十四 APPLE時鐘

電腦的一項特色就是能夠以其變化萬千的軟體程式，來做各種巧妙的應用，只要改變軟體程式，則電腦的所執行的工作馬上做十萬八千里的轉變。這種特色，尤其是在以即時（ REAL TIME ）處理見長的微電腦，更顯出其優點。

本程式就是利用 APPLE Ⅱ 來執行時鐘的功能，在畫面上以數字顯示出時間，由喇叭發出秒針的嘀嗒聲，同時又具有鬧鐘報時的功能，使用上與一般定時鬧鐘無異。而其最大優點就是程式設計者可加以發展而透過APPLE的 I/O輸出入埠來對外界做時序的控制，這便是電腦爲其它控制方式所不及之處，程式完全以 BASIC 寫出，如果考慮到控制速度方面的問題時，加上一段機械語言程式也不是什麼困難的事情吧！

程式例

```
]LIST

1 DE = 100: DIM N(12),S(7),A$(9),TT
     (127):N1 = 1:N2 = 2:N3 = 1982:
     Q = 1
3  TEXT : GOSUB 9100: HOME : VTAB (
     2): HTAB (17): INVERSE : PRINT
     "TIMER": NORMAL
4  HTAB (9): PRINT "CLOCK/CALENDAR"
```

```
    : VTAB (6): HTAB (5): PRINT "O
    PTIONS:": HTAB (14)
5  POKE 32,13: PRINT "1-SET DATE": PRINT
    "2-SET TIME": PRINT "3-SET  TI
    CK": PRINT "4-SET ALARM": PRINT
    "5-START CLOCK": POKE 32,0
6  PRINT : PRINT "TIME KEEPERS:": HTAB
    (5): PRINT "TO START ONE, HIT
    A NUMBER WHILE": PRINT "THE CL
    OCK IS RUNNING.": HTAB (5)
7  PRINT "TO FIND OUT THE ELASPED T
    IME, TYPE": PRINT "THE SAME NU
    MBER AGAIN."
8  POKE 32,0: POKE 33,39: POKE 34,1
    8: POKE 35,23: HOME : INPUT "W
    HICH OPTION";O: IF O < 1 OR O >
    5 THEN 8
9  IF O = 1 THEN 25
10  IF O = 2 THEN 90
11  IF O = 3 THEN 95
12  IF O = 5 THEN 97
13  INPUT "ALARM TIME (H, M, S)";H1
    ,M1,Z1: IF H1 >  - 1 AND H1 <
    24 AND M1 >  - 1 AND M1 < 60 AND
    Z1 >  - 1 AND Z1 < 60 THEN 8
15  PRINT "ILLEGAL TIME (24 HOUR FO
    RMAT)!": GOTO 113
25  INPUT "CURRENT DATE (M, D, Y)";
    N1,N2,N3
28  IF N3 < 100 THEN N3 = 1900 + N3

30 N(1) = 31:N(2) = 29:N(3) = 31:N(
```

```
     4) = 30:N(5) = 31:N(6) = 30:N(
     7) = 31:N(8) = 31:N(9) = 30:N(
     10) = 31:N(11) = 30:N(12) = 31

40  IF N2 > 0 AND N2 < N(N1) + 1 THEN
     50
41  GOSUB 8000: PRINT A$;" ONLY HAS
      A MAX OF ";N(N1);" DAYS!": GOTO
     25
50  IF N1 <  > 2 OR N2 <  > 29 THEN
     60
51  IF N3 = 4 *  INT (N3 / 4) THEN
     60
52  PRINT "THE NEXT LEAP YEAR IS ";
     4 +  INT (N3 / 4) * 4;"!": GOTO
     25
60  GOTO 8
90  INPUT "CURRENT TIME (H, M, S) "
     ;H,M,Z
91  IF H >  - 1 AND H < 24 AND M >
      - 1 AND M < 60 AND Z >  - 1 AND
     Z < 60 THEN 8
92  PRINT "ILLEGAL TIME (24 HOUR FO
     RMAT)!": GOTO 90
95  INPUT "SOUND OF TICK?";A$: IF A
     $ = "N" THEN Q = 0
96  GOTO 8
97  FOR I = 1 TO 127:TT(I) = 0: TEXT
     : HOME : GR
98 LY = 15: GOSUB 9000
100 LOCX = 1:NUM = H: GOSUB 7000
110 LOCX = 15:NUM = M: GOSUB 7000
```

```
120 LOCX = 29:NUM = Z: GOSUB 7000
150 Z = Z + 1: IF Z < 60 THEN 500
160 Z = 0
170 M = M + 1: IF M < 60 THEN 500
180 M = 0:H = H + 1
190  IF H < 24 THEN 500
200 H = 0: GOSUB 10000
500  IF Q = 0 THEN 502
501 K =  PEEK ( - 16336) -  PEEK ( -
     16336) +  PEEK ( - 16336)
502  IF H <  > H1 OR M <  > M1 OR Z
      - Z1 < 1 OR Z - Z1 > 10 THEN
     508
503  GOSUB 13000: GOTO 510
508  FOR K = 1 TO DE: NEXT K
509  GOSUB 11100
510  IF Z THEN 120
520  IF M THEN 110
530  GOTO 100
1000  REM  SUBROUTINE DIGIT (DIG,LO
     CX,LOCY)
1010  FOR K = 1 TO 7:S(K) = 1: NEXT
     K
1020  ON DIG + 1 GOTO 1050,1100,115
     0,1200,1250,1300,1350,1400,145
     0,1500
1050 S(7) = 0
1060  GOSUB 5000: RETURN
1100 S(1) = 0:S(4) = 0:S(5) = 0:S(6
     ) = 0:S(7) = 0
1110  GOSUB 5000: RETURN
1150 S(2) = 0:S(5)    0
```

```
1160  GOSUB 5000: RETURN
1200 S(5) = 0:S(6) = 0
1210  GOSUB 5000: RETURN
1250 S(1) = 0:S(4) = 0:S(6) = 0
1260  GOSUB 5000: RETURN
1300 S(3) = 0:S(6) = 0
1310  GOSUB 5000: RETURN
1350 S(3) = 0:S(4) = 0
1360  GOSUB 5000: RETURN
1400 S(1) = 0:S(5) = 0:S(6) = 0:S(7
     ) = 0
1410  GOSUB 5000: RETURN
1450  GOSUB 5000: RETURN
1500 S(1) = 0:S(6) = 0
1510  GOSUB 5000: RETURN
5000  COLOR= 0
5010  FOR K = 1 TO 7
5020  IF  NOT S(K) THEN  GOSUB 6000

5030  NEXT K
5050  FOR K = 1 TO 7
5060  IF  NOT S(K) THEN 5090
5070  COLOR= 9
5080  GOSUB 6000
5090  NEXT K
5100  RETURN
6000  ON K GOTO 6100,6200,6300,6400
     ,6500,6600,6700
6100  HLIN LOCX,LOCX + 4 AT LY + 20
     : RETURN
6200  VLIN LY + 10,LY + 20 AT LOCX +
     4: RETURN
```

```
6300  VLIN LY,LY + 10 AT LOCX + 4: RETURN

6400  HLIN LOCX,LOCX + 4 AT LY: RETURN

6500  VLIN LY,LY + 10 AT LOCX: RETURN

6600  VLIN LY + 10,LY + 20 AT LOCX:
      RETURN
6700  HLIN LOCX,LOCX + 4 AT LY + 10
     : RETURN
7000 DIG =  INT (NUM / 10): GOSUB 1
     000
7010 LOCX = LOCX + 6
7020 DIG = NUM - 10 * DIG: GOSUB 10
     00
7030  RETURN
8000  ON N1 GOTO 8010,8020,8030,804
     0,8050,8060,8070,8080,8090,810
     0,8110,8120
8010 A$ = "JANUARY": RETURN
8020 A$ = "FEBRUARY": RETURN
8030 A$ = "MARCH": RETURN
8040 A$ = "APRIL": RETURN
8050 A$ = "MAY": RETURN
8060 A$ = "JUNE": RETURN
8070 A$ = "JULY": RETURN
8080 A$ = "AUGUST": RETURN
8090 A$ = "SEPTEMBER": RETURN
8100 A$ = "OCTORBER": RETURN
8110 A$ = "NOVEMBER": RETURN
8120 A$ = "DECEMBER": RETURN
9000  VTAB (22): GOSUB 8000: CALL
```

```
936: HTAB (17 -  LEN (A$) / 2)
    : PRINT A$;" ";N2;", ";N3: RETURN

9100  FOR I = 0 TO 20: READ A: POKE
    768 + I,A: NEXT I
9101  DATA 173,48,192,136,208,5,206
    ,22,3,240,9,202,208,245,174,21
    ,3,76,0,3,96
9102  RETURN
10000 N2 = N2 + 1: IF N2 > N(N1) OR
    (N2 = 29 AND N3 <  >  INT (N3 /
    4) * 4) THEN 11000
10100  GOSUB 9000: RETURN
11000 N1 = N1 + 1:N2 = 1: IF N1 < 1
    3 THEN 10100:N1 = 1:N3 = N3 +
    1: GOTO 10100
11100 K =  PEEK ( - 16384): IF K <
    128 THEN  RETURN
11101  POKE  - 16368,0:K = K - 128
11200  IF TT(K) < 0 THEN 11300: PRINT
    "";: RETURN
11300 TT(K) = 1: PRINT "";: RETURN


12000  POKE 789,S1: POKE 790,S2: CALL
    768: RETURN
13000  FOR S1 = 100 TO 110:S2 = 5: GOSUB
    12000: NEXT S1: RETURN
```

程式説明

程式由第1行設定變數DE、N1、N2、N3、Q的值以及陣列(ARRAY)N、S、A$、TT等的維數(DIMENSION),接着至副程

式9100將發聲副程式讀入記憶位址＄0300至＄0314（即768至788），至於接下來由第3行至第8行將程式MANU印出並且要求操作者鍵入所選擇的項目（1至5），而9至13行則把鍵入的號碼分別跳至對應的行號。

第13行是設定鬧鈴時間，其變數爲H1、M1、Z1，第25行是設定日期N1、N2、N3，其中N3若小於100則自動加上1900成爲西元（當然指的是將1982簡寫成82的情形，若是民國則程式應改成加上1911），30行設定一年12個月的日數，如果操作者輸入錯誤則由第40、41行及8000副程式來糾正，50至52行找出閏年。90行輸入時間H、M、Z（時、分、秒），第95行詢問是否發出鐘擺的嘀嗒聲。

第97行就開始了時鐘的運轉。其中6000行至6700行的副程式是畫出8字的7個筆劃，如下圖：

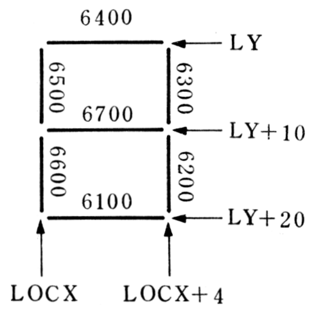


圖 2-16

而這7個筆劃是由1020至1500行來決定畫出何種數字，譬如“3”這個數字是由第1200行設定S(5)＝0及S(6)＝0，然後GOSUB 5000先設定COLOR＝0爲黑色，用IF NOT S(K)將這兩畫由先前的字消去，再由5050行設定COLOR＝9爲橙色，將其餘的筆劃畫出就成爲一個“3”字了。至於時、分、秒在螢幕上的位置是由100行至120行來定出LOCX的值就可決定其水平方向的位置。時、分、秒本身以

及秒至分、分至時的進位是由150行至第200行來完成，而由7000至7020行來設定畫出在螢幕上。鬧鐘是由502行來監視，在到達預定時間至以後10 秒之間，程式至 13000 行叫出聲音副程式，這時喇叭發出咕嚕咕嚕的聲音。

如果發現鐘的速度太快的話（據筆者試驗是太快），可適當加大508行的FOR－NEXT迴路的DE值調整之。充份利用程式的彈性，可設定多重定時，或利用APPLE的I/O來做各種控制，應用上應該是很方便的。

程式十五 太空防衛戰

這是屬於TV GAME一類的遊戲，利用APPLE的高解像畫面(HIRES)以及圖形表(SHAPE TABLE請參考程式三有關圖形表的使用)來模擬一個太空戰爭的情形。由於程式是以APPLESOFT BASIC來寫的，因此看起來有點慢。如果將太空船軌道以及砲擊的程式用機械語言來寫的話，就可有很大的改善。

遊戲進行時，在畫面上出現防衛的太空船沿着一軌道路線在移動着，接着螢幕上打出"入侵者到來"的警示，入侵者亦是沿着軌道在螢幕上繞行，軌道有兩條，入侵者及防衛者可任意在其中之一。防衛者有4個飛彈可用以對入侵開火，開火的開關是由APPLE的GAME PADDLE 0來控制的，飛彈發射後在一段時間內可以再補充，所以防衛者具有無限的火力供應。

當入侵者接近防衛者至一段距離時就會自動開火，入侵者的武器是由太空船發射的死光，但是只在一有限的距離有效。當雙方之一被擊中五次之後，遊戲即告結束(入侵者的攻擊並不是每次皆命中，但却能消耗防衛者的能量，當防衛者能量爲零時，就算摧毀了)。

程 式

```
]LIST

1  REM  **  ORBITAL WAR GAME
```

```
2  GOSUB 1600
5  GOTO 1000
10  FOR J = 0 TO 1: HCOLOR= 0: DRAW
    J + 1 AT VS(J,4),VS(J,5): HCOLOR=
     ABS (VS(J,0)): DRAW J + 1 AT
    VS(J,6),VS(J,7):VS(J,2) = VS(J
    ,2) + VS(J,3)
20 VS(J,4) = VS(J,6):VS(J,5) = VS(J
    ,7):VS(J,6) =  COS (VS(J,2)) *
    VS(J,9) + VS(J,11):VS(J,7) =  SIN
    (VS(J,2)) * VS(J,10) + VS(J,12
    ): IF VS(J,0) >  - 1 THEN 40
30  HCOLOR= 0: FOR K = 0 TO VS(J,16
    ): DRAW 4 - J AT VS(J,4) - SD(
    3 - J,1,0) +  RND (1) * 7,VS(J
    ,5) - SD(3 - J,1,1) +  RND (1)
     * 9: NEXT :VS(J,16) = VS(J,16
    ) + .47
35  FOR D = 1 TO 180: NEXT D
40  NEXT :VS(1,15) =  SQR (((VS(0,4
    ) + SD(0,1,0)) - (VS(1,4) + SD
    (1,1,0))) ^ 2 + ((VS(0,5) + SD
    (0,1,1)) - (VS(1,5) + SD(1,1,1
    ))) ^ 2): RETURN
50  FOR K = 0 TO 3: IF  NOT W(K,0) THEN
    70
60  NEXT : GOTO 100
70 W(K,3) =  COS (VS(0,15) - C(0,6)
    ) * SD(0,0,0) + VS(0,6) + SD(0
    ,1,0): IF W(K,3) < C(0,3) OR W
    (K,3) > C(0,4) THEN W(K,3) = C
    (0,4): GOTO 100
```

```
80 W(K,4) =  SIN (VS(0,5) - C(0,6))
      * SD(0,0,1) + VS(0,7) + SD(0,
     1,0): IF W(K,4) < C(1,3) OR W(
     K,4) > C(1,4) THEN W(K,3) = C(
     0,4):W(K,4) = C(1,4): GOTO 100

90 W(K,0) = 1:W(K,5) =  COS (VS(0,1
     5) - C(0,6)) * 18:W(K,6) =  SIN
     (VS(0,15) - C(0,6)) * 8:VS(0,1
     4) = VS(0,14) - VS(0,13)
100  FOR J = 0 TO 3: HCOLOR= 0: DRAW
     3 AT W(J,1),W(J,2): HCOLOR= W(
     J,0): DRAW 3 AT W(J,3),W(J,4):
     W(J,1) = W(J,3):W(J,2) = W(J,4
     ):W(J,3) = W(J,3) + W(J,5):W(J
     ,4) = W(J,4) + W(J,6)
110  IF W(J,3) < C(0,3) OR W(J,3) >
     C(0,4) OR W(J,4) < C(1,3) OR W
     (J,4) > C(1,4) THEN W(J,0) = 0
     :W(J,3) = C(0,4):W(J,4) = C(1,
     5):W(J,5) = 0:W(J,6) = 0
120  NEXT : RETURN
140  FOR J = 0 TO 3: IF W(J,0) = 0 OR
     VS(1,0) <  > 1 THEN 170
150  IF  ABS (VS(1,4) + SD(1,1,0) -
     W(J,1) - SD(2,1,0)) < 9 AND  ABS
     (VS(1,5) + SD(1,1,1) - W(J,2) -
     SD(2,1,1)) < 5 THEN VS(1,0) =
      - 1: VTAB 24: HTAB 1: PRINT "
     INTRUDER DESTROYED";:VS(1,16) =
     0:DC = DC + 1: GOTO 475
170  NEXT :VS(0,15) =  PDL (0) * C(
```

```
       0,5):VS(0,14) = VS(0,14) + VS(
       0,8): VTAB 23: HTAB 17: PRINT
        INT (VS(0,15) * C(1,5) + .5);
        SPC( 3);: RETURN
390  HCOLOR= 0: DRAW 4 AT W(4,1),W(
       4,2): IF VS(1,15) > VS(1,14) OR
       VS(1,0) < 1 THEN W(4,3) = C(0,
       4):W(4,4) = C(1,4): FOR J = 0 TO
       300: NEXT : RETURN
400 VS(1,8) = (1 - VS(1,15) < VS(1,
       13)) * ((VS(1,15) - VS(1,13)) /
       (VS(1,14) - VS(1,13))):W(4,5) =
       VS(1,6) - VS(0.6): IF  ABS (W(
       4,5)) < .001 THEN W(4,5) = C(0
       ,6) + (VS(1,7) > VS(0,7)) * PI
       : GOTO 420
410 W(4,5) =  ATN ((VS(1,7) - VS(0,
       7)) / W(4,5)) + (VS(1,6) > VS(
       0,6)) * PI
420 W(4,5) = W(4,5) + ( - 1) ^  INT
       ( RND (1) + .5) *  RND (1) * V
       S(1,8) * PI / 5:W(4,6) = VS(1,
       15) + ( - 1) ^  INT ( RND (1) +
       .5) *  RND (1) * VS(1,8) * (VS
       (1,14) - VS(1,15))
430 W(4,3) = VS(1,6) +  COS (W(4,5)
       ) * W(4,6):W(4,4) = VS(1,7) +
        SIN (W(4,5)) * W(4,6): IF W(4
       ,3) < C(0,3) OR W(4,3) > C(0,4
       ) OR W(4,4) < C(1,3) OR W(4,4)
        > C(1,4) THEN  FOR J = 0 TO 5
       5: NEXT J: RETURN
```

```
440  FOR J = 1 TO 6: HCOLOR= J -  INT
     (J / 2) * 2: HPLOT VS(1,6) + S
     D(2,0,0),VS(1,7) + SD(2,0,1) TO
     W(4,3),W(4,4): NEXT : HCOLOR=
     1: DRAW 4 AT W(4,3),W(4,4):W(4
     ,1) = W(4,3):W(4,2) = W(4,4)
450 VS(1,8) =  SQR ((W(4,3) - (VS(0
     ,6) + SD(0,1,0))) ^ 2 + (W(4,4
     ) - (VS(0,7) + SD(0,1,1))) ^ 2
     )
460 VS(0,14) = VS(0,14) - (VS(1,14)
      -  INT (VS(1,8))) * 6: IF VS(
     0,14) > 0 THEN 480
470  VTAB 24: HTAB 1: PRINT "DEFEND
     ER DESRROYED ";:VS(0,0) =  - 1
     :VS(0,16) = 0:VS(0,14) = 0:VS(
     0,8) = 0:IC = IC + 1
475  HCOLOR= 0: DRAW 4 AT W(4,1),W(
     4,2): RETURN
480 W(4,3) = C(0,4):W(4,4) = C(1,4)
     : RETURN
610  GOSUB 140
620  GOSUB 10: VTAB 22: HTAB 15: PRINT
     VS(0,14); SPC( 2);: HTAB 34: PRINT
      INT (VS(1,15) + .5); SPC( 3);

640  IF VS(0,0) + VS(1,0) = 2 AND  PEEK
     ( - 16287) > 127 AND VS(0,14) >
     VS(0,13) THEN  GOSUB 50: GOTO
     740
660  GOTO 100
740  GOSUB 140: IF VS(0,0) + VS(1,0
```

```
     ) = 2 THEN  GOSUB 390
800  IF VS(0,16) < 5 AND VS(1,16) <
     5 THEN 610
840  IF VS(0,0) < 0 THEN VS(0,0) =
     0
850  IF VS(1,0) < 0 THEN VS(1,0) =
     0
860  FOR K = 0 TO 7: GOSUB 10: FOR
     D = 1 TO 500: NEXT D,K
998  RETURN
1000  DIM VS(1,16),W(4,7): POKE 232
     ,0: POKE 233,3:J = 0: SCALE= 1
     : ROT= 0:K = 0
1010  DIM C(1,6),SD(3,1,1),OV(1):PI
      = 3.14159
1020  FOR J = 0 TO 3: FOR K = 0 TO
     1: FOR L = 0 TO 1: READ SD(J,K
     ,L): NEXT L,K,J
1030  FOR K = 0 TO 1: FOR J = 0 TO
     4: READ C(K,J): NEXT J,K: READ
     C(0,5),C(1,5)
1040 OV(1) = 5 / 57.2958:OV(0) = OV
     (1) / 1.5435
1050  GOSUB 3000
1070 IC = 0:DC = 0:VS(0,16) = 0:VS(
     1,16) = 0:C(0,6) = 1.5707
1080 VS(0,0) = 1:VS(1,0) = 0:VS(0,1
     ) =  INT ( RND (1) + .5):VS(1,
     1) =  INT ( RND (1) + .5):VS(0
     ,2) =  RND (2 * PI):VS(1,2) =
     VS(0,2) + PI + ( - 1) ^ ( INT
     ( RND (1) + .5)) *  RND (PI)
```

```
1090 VS(0,3) = OV(VS(0,1)) * (1 - 2
      * INT ( RND (1) + .5)):VS(1,
     3) = OV(VS(1,1)) * (1 - 2 * INT
     ( RND (1) + .5))
1100  FOR J = 0 TO 4:W(J,0) = 0:W(J
     ,3) = C(0,4):W(J,4) = C(1,4):W
     (J,5) = 0:W(J,6) = 0:W(J,1) =
     W(J,3):W(J,2) = W(J,4): NEXT
1110  FOR K = 0 TO 1:VS(K,9) = C(VS
     (K,1),0) - SD(K,0,0) / 2:VS(K,
     10) = C(VS(K,1),1) - SD(K,0,1)
      / 2:VS(K,11) = C(0,2) - SD(K,
     1,0):VS(K,12) = C(1,2) - SD(K,
     1,1): NEXT K
1120  FOR K = 0 TO 1:VS(K,4) =  COS
     (VS(K,2)) * VS(K,9) + VS(K,11)
     :VS(K,6) = VS(K,4):VS(K,5) =  SIN
     (VS(K,2)) * VS(K,10) + VS(K,12
     ):VS(K,7) = VS(K,5): NEXT
1130 VS(0,8) = 85:VS(0,13) = 134:VS
     (1,13) = 40:VS(1,14) = 120
1140  HGR : VTAB 21: HOME : VTAB 21
     : PRINT "DEFFENDER = ";DC;: HTAB
     23: PRINT "INTRUDER = ";IC
1150  PRINT "ENERGY LEVEL ="; SPC(
     8);"DISTANCE ="
1160  PRINT "FIRE DIRECTION ="
1170  HCOLOR= 1: HPLOT 1,0 TO 279,0
      TO 279,159 TO 1,159 TO 1,0: DRAW
     1 AT VS(0,4),VS(0,5):VS(0,16) =
     0:VS(1,16) = 0
1180  FOR K = 0 TO 8: GOSUB 10: FOR
```

```
     D = 1 TO 550: NEXT D,K: VTAB 2
     4: FLASH : PRINT "WARNING!! IN
     TRUDER APPROCHING";
1190  NORMAL : FOR K = 0 TO 4: GOSUB
     10: FOR D = 1 TO 550: NEXT D,K
     : VTAB 23: CALL  - 958:VS(1,0)
      = 1:VS(0,14) = 3000
1200  GOSUB 610: IF IC < 5 AND DC <
     5 THEN 1080
1210  VTAB 23: HTAB 24: INPUT "TRY
     AGAIN ?";Q$: IF Q$ <  > "N" THEN
     1070
1220  TEXT : HOME : END
1600  FOR K = 768 TO 1000: READ J: IF
     J =  - 1 THEN 1840
1610  POKE K,J: NEXT K: GOTO 1840
1620  DATA 4,0,10,0,105,0,189,0,202
     ,0
1630  DATA 45,5,45,5,45,5,45,5,73,5
     8,7
1640  DATA 63,7,63,7,63,7,63,7,63,5
     ,63,7
1650  DATA 191,73,73,105,5,45,5,141
     ,251,63,7,255
1660  DATA 74,45,5,109,21,7,59,7,7,
     255,31,7
1670  DATA 74,45,5,7,45,5,45,5,13,5
     ,250,63
1680  DATA  7,63,7,7,255,31,7,155,4
     1,5,45,5
1690  DATA 45,5.45,5,45,5,45,5,45,5
     ,218,59
```

```
1700  DATA  7,63,7,63,7,63,7,7,63,7
      ,7,0
1710  DATA 77,9,5,73,77,77,58,7,63,
      7,63
1720  DATA 7,31,7,63,7,63,7,159,45,
      5,45,5
1730  DATA 45,5,45,5,45,5,45,7,7,5,
      45,5
1740  DATA 45,5,218,59,7,63,5,63,7,
      63,7,15
1750  DATA 63,7,63,7,5,63,25,191,73
      ,13,5,45
1760  DATA 5,45,5,5,45,5,209,31,7,6
      3,7,63
1770  DATA 7,63,7,159,41,5,105,41,5
      ,73,5,45,0
1780  DATA 77,250,31,7,155,9,5,105,
      77,218,27
1790  DATA 7,0
1800  DATA 77,109,58,7,63,7,255,87,
      41,5,5
1810  DATA 9,5,13,5,218,63,7,5,63,7
      ,255,87
1820  DATA 9,5,9,5,5,0
1830  DATA -1
1840  RETURN
2100  DATA  10,10,4,2
2110  DATA 11,7,3,3
2120  DATA 3,4,0,1
2130  DATA 5,5,1,2
2140  DATA 128,75,138,5,270
2150  DATA 92,60,80,5,150
```

```
2160  DATA  .02464,57.2958
3000  HOME : VTAB 8: HTAB 8: PRINT
      "===ORBITAL WAR GAME===": PRINT
      : PRINT : HTAB 8: PRINT "== BA
      SIC    PROGRAM =="
3030  FOR J = 1 TO 1500: NEXT J: RETURN
```

程式説明

程式開始先至1600副程式將圖形表的資料（1620行至1820行）以第1610行讀入至記憶空間的第三頁區（＄0300＝768），接着至第1000行設定圖形表之初值，其中POKE 232,0以及POKE 233,3是將圖形表指位器（POINTER）的內含值存入＄0300的值，亦即指示圖形表是由＄0300位址開始的。由1010行至1040行是設定一些數值，譬如SCALE＝1，ROT ＝ 0以及表示軌道的速度OV(1)、OV(2)等。1500行的GOSUB 3000是將遊戲名稱抬頭印出在螢幕上。接着1070至1130安排畫面的各種相關資料，包含了位置、角度、速度、能量、武器資料等等。

遊戲眞正開始是由第610行至850行之間，首先由GOSUB 140查看防衛者的攻擊是否將入侵者擊中？接着GOSUB 10來移動畫面的太空船，並算出二者之距離。回到640主程式後，接着判斷是否攻擊的按鈕（PADDLE）按下，如果已按下而且能量足夠的話則至第 50 行發射飛彈攻擊。740行的GOSUB 140用來判斷攻擊是否成功，若不成功則程式轉至390行計算兩者的相關角度、距離、運動情況以便接着由入侵者發動攻擊，第440副程式執行入侵者的攻擊，如果成功則印出“防衛者被摧毀（DEFENDER DESTROYED）”的字幕。當有一方的太空船被擊中五次時，程式卽告結束。本程式用到的變數很多，而且多用陣列（ARRAY）表示，簡列如下：

DIM C(1,6)

C(A,B)

A＝0,1＝軌道代號

B＝0,1 0表軌道X範圍，1表軌道Y範圍

C(I,J)

I＝0,1 0表X軸值，1表Y軸值

J＝2～5 C(I,3)表I所代表的軸之最小值，C(I,4)表最大值，C(0,5)是將控制器(PADDLE)的值(0至255)轉換成(0至π)，C(1,5)則將徑度量轉換成角度，C(0,6)將控制器的值轉換成螢幕上方向的角度，即東－90度、西－270度、南－180度、北－360度，C(1,6)記錄畫出太空船摧毀的次數。

DC＝防衞者被擊中次數

IC＝入侵者被擊中次數

OV(0)＝第0號軌道之角速度

OV(1)＝第1號軌道之角速度

DIM SD(3,1,1)＝圖形資料

SD(A,B,C)

A=0～3 0＝防衞者，1＝入侵者，2＝防衞者飛彈，3＝入侵者死光。

B＝0,1 0＝範圍，1＝太空船圖形中心至原點座標(0,0)之距離。

C＝0,1 0＝X軸之值，1＝Y軸之值。

DIM VS(1,15)＝空間之各種狀態

VS(A,B)

A＝0，1　0＝防衛者，1＝入侵者

B＝0～15　0＝該太空船之狀態，譬如VS(1，0)＝0表示入侵者不存在，VS(1，0)＝1表示存在，VS(1，0)＝－1表示入侵者被擊中。1＝軌道號碼，譬如VS(0，1)＝0表示防衛者在外軌道，VS(1，1)＝1表示入侵者在內軌道。2＝角度值。3＝軌道速度。4＝X軸之值。5＝Y軸之值。6＝下個X軸值。7＝下個Y軸值。VS(0，8)＝所增加的能量。VS(1，8)＝入侵者打擊防衛的精確度或距離。9＝X的範圍，10＝Y的範圍。11＝至螢幕原點的X方向距離。12＝至螢幕原點的Y方向距離。VS(0，13)＝防衛者能量基準。VS(0，14)＝防衛者能量。VS(1，13)＝入侵者射程基準。VS(1，14)＝入侵者最大射程。VS(0，15)＝防衛者射擊方向。VS(1，15)＝二者距離。16＝擊中次數。

DIM　W(4，6)＝武器狀態

A＝0，1，2，3＝防衛者的4枚飛彈

A＝4　表示在某個距離內，入侵者可發射無限次數的死光

B＝0～4　0＝顏色。1＝X位置之值。2＝Y位置之值。3＝下個X位置之值。4＝下個Y位置之值(以上係對二者武器而言)。

B＝5，6　5＝入侵者武器X值之亂數。6＝入侵者武器Y值之亂數。

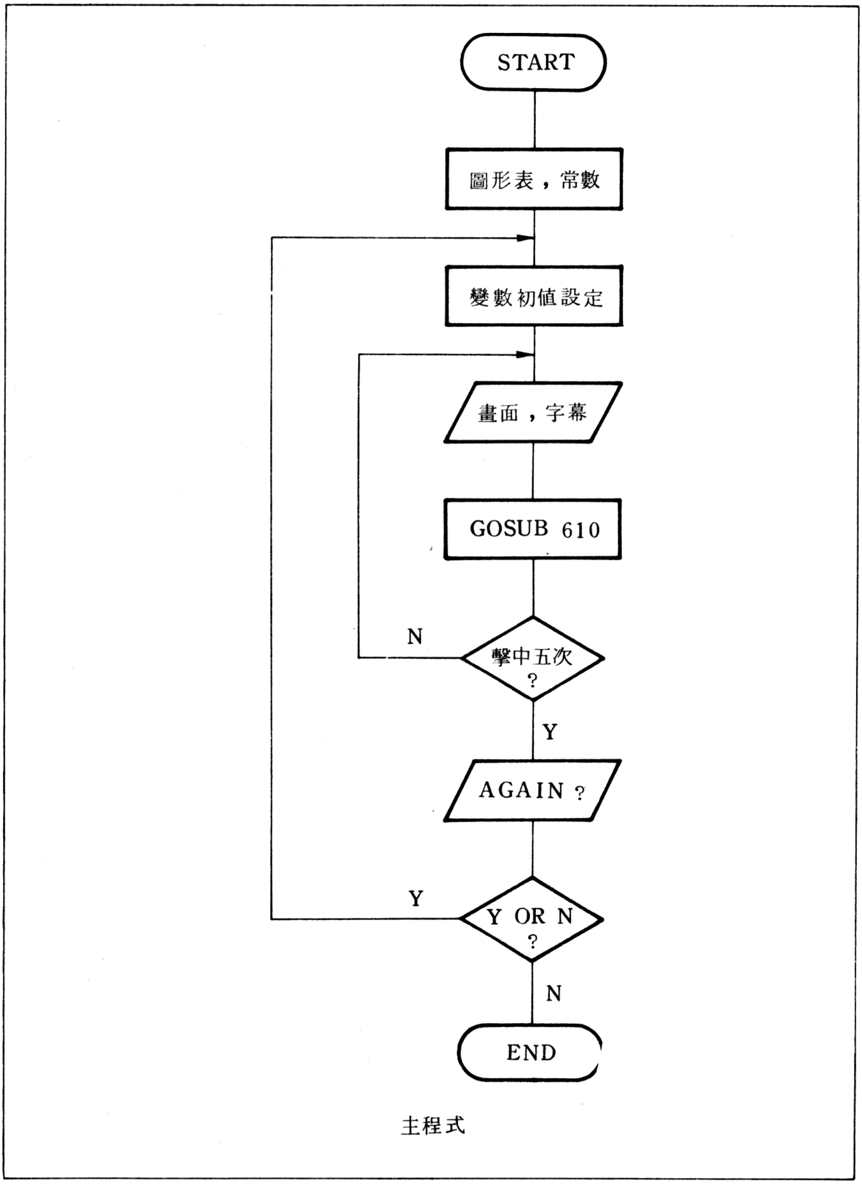
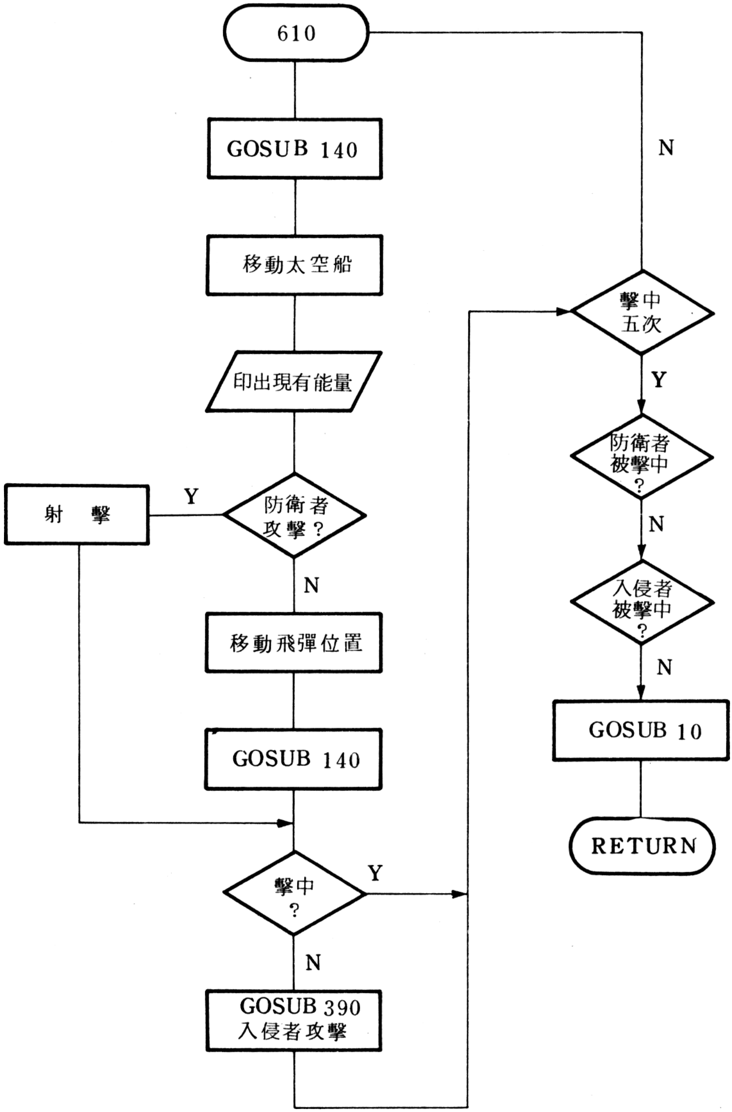
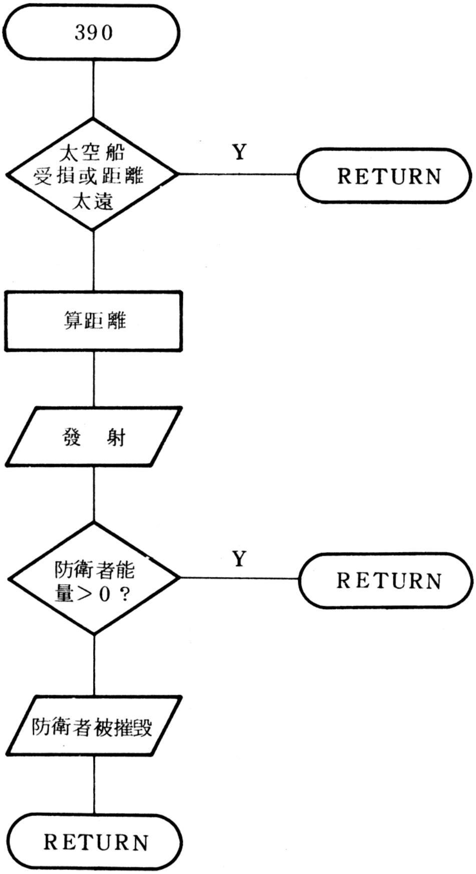

START
圖形表，常數
變數初值設定
畫面，字幕
GOSUB 610
N
擊中五次
?
Y
AGAIN ?
Y
Y OR N
?
N
END
主程式

圖 2-17 (a)



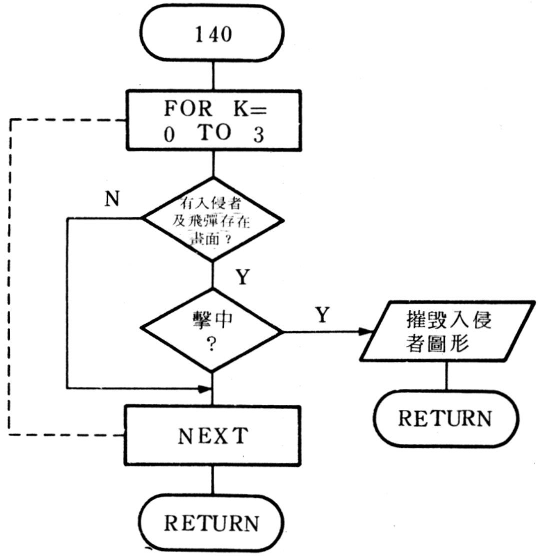
610副程式

圖 2-17(b)



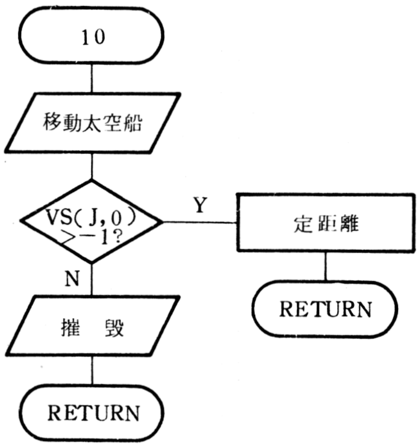
390 副程式

圖 2-17(c)



140副程式

圖 2-17(d)



10 副程式

圖 2-17(e)

附錄〔A〕 十六進位 二進位與十六進位 十進位轉換表

十六進位 二進位之轉換

十六進位與二進位之間的轉換是十六進位數的一個位數相對於二進位數的四個位數來做轉換的，如果二進位數由右向左每次四個位數成一組變換之後，所剩不足四個位數則在左邊加上0來補足，下列便是兩個例子：

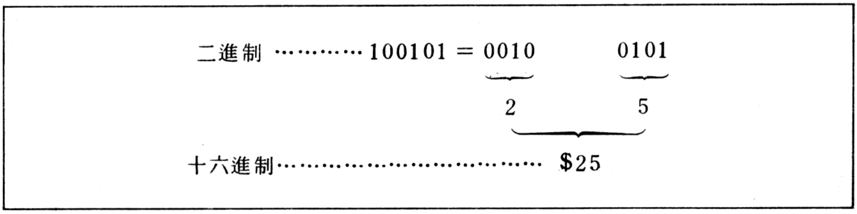


圖 A-1 二進制至十六進制的轉換

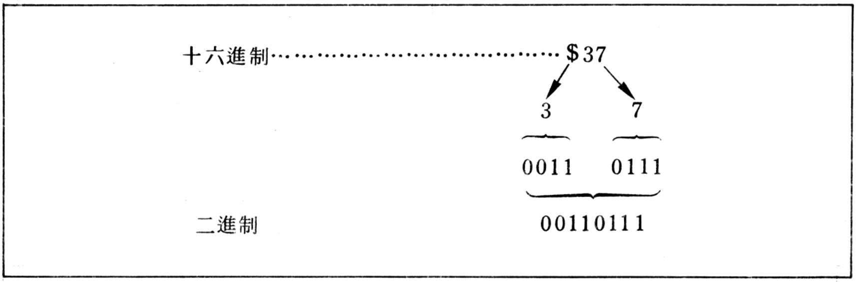


圖 A-2 十六進制至二進制的轉換

如果要將十進制的數與十六進制相互轉換則依下兩例進行。

十 六 進 位	二 進 位
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

圖A-3 十六進制——二進制互換表

$$\$16E9 = 9\times16^0 + E\times16^1 + 6\times16^2 + 1\times16^3$$

$$= 9 + 14\times16 + 6\times256 + 4096$$

$$= 5865$$

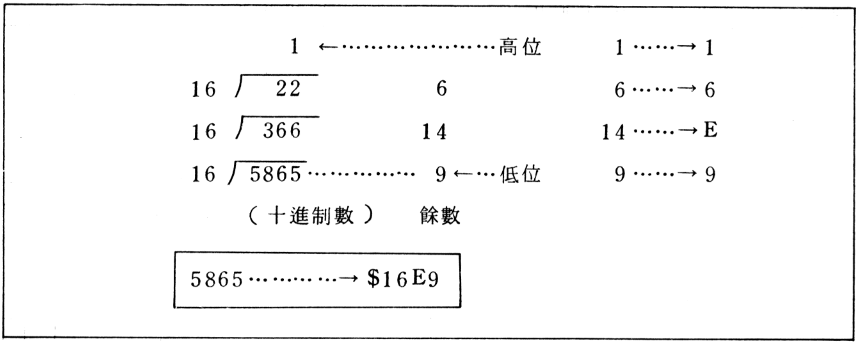
4096 → 1

256 → 6

16 → E

1 → 9

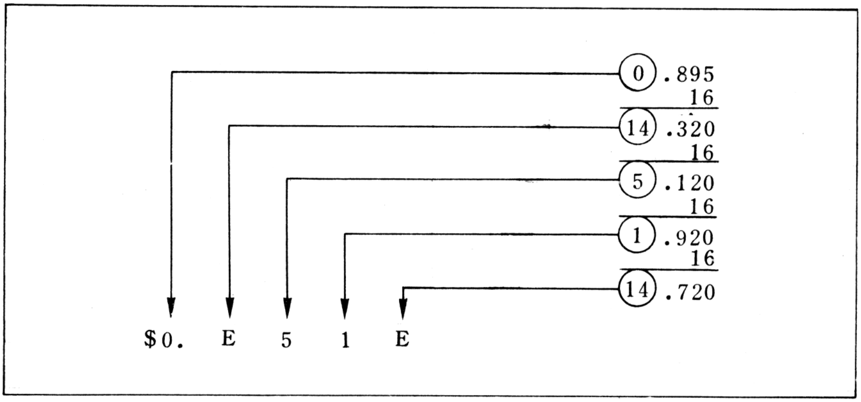
圖A-4 十進制至十六進制的轉換



圖A-5　十進制至十六進制的轉換

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010	0011	0012	0013	0014	0015
01	0016	0017	0018	0019	0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	0030	0031
02	0032	0033	0034	0035	0036	0037	0038	0039	0040	0041	0042	0043	0044	0045	0046	0047
03	0048	0049	0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	0060	0061	0062	0063
04	0064	0065	0066	0067	0068	0069	0070	0071	0072	0073	0074	0075	0076	0077	0078	0079
05	0080	0081	0082	0083	0084	0085	0086	0087	0088	0089	0090	0091	0092	0093	0094	0095
06	0096	0097	0098	0099	0100	0101	0102	0103	0104	0105	0106	0107	0108	0109	0110	0111
07	0112	0113	0114	0115	0116	0117	0118	0119	0120	0121	0122	0123	0124	0125	0126	0127
08	0128	0129	0130	0131	0132	0133	0134	0135	0136	0137	0138	0139	0140	0141	0142	0143
09	0144	0145	0146	0147	0148	0149	0150	0151	0152	0153	0154	0155	0156	0157	0158	0159
0A	0160	0161	0162	0163	0164	0165	0166	0167	0168	0169	0170	0171	0172	0173	0174	0175
0B	0176	0177	0178	0179	0180	0181	0182	0183	0184	0185	0186	0187	0188	0189	0190	0191
0C	0192	0193	0194	0195	0196	0197	0198	0199	0200	0201	0202	0203	0204	0205	0206	0207
0D	0208	0209	0210	0211	0212	0213	0214	0215	0216	0217	0218	0219	0220	0221	0222	0223
0E	0224	0225	0226	0227	0228	0229	0230	0231	0232	0233	0234	0235	0236	0237	0238	0239
0F	0240	0241	0242	0243	0244	0245	0246	0247	0248	0249	0250	0251	0252	0253	0254	0255

圖A-6　十六進制——十進制互換表



圖A-7　十進制小數 —— 十六進制小數

如果是具有小數點的數，則依下兩例轉換之。

1. 將16進制的小數化成整數並以乘上 16^{-n} ，
   n 即爲原來小數之位數。
   $0.E51E = $E51E × 16^{-4}
2. 將整數部份化成10進制
   $E51E = E × 4096 + 5 × 256 + 1 ×
   16 + E = 58654
3. 再將指數部份乘回來
   $E51E × 16^{-4} = 58654 × 16^{-4}
   = 0.8915
   所求出來的值 0.8915 與前一個例子裏的值 0.895 的差別是因爲前一個例子裏有 0.72 的值被忽略了。

圖 A-8 十六進制小數 ── 十進制小數

附錄〔B〕 ASCII碼

表B-1 APPLE的ASCII碼

Key	Alone	CTRL	SHIFT	Both	Key	Alone	CTRL	SHIFT	Both
space	$AØ	$AØ	$AØ	$AØ	RETURN	$8D	$8D	$8D	$8D
Ø	$BØ	$BØ	$BØ	$BØ	G	$C7	$87	$C7	$87
1!	$B1	$B1	$A1	$A1	H	$C8	$88	$C8	$88
2"	$B2	$B2	$A2	$A2	I	$C9	$89	$C9	$89
3#	$B3	$B3	$A3	$A3	J	$CA	$8A	$CA	$8A
4$	$B4	$B4	$A4	$A4	K	$CB	$8B	$CB	$8B
5%	$B5	$B5	$A5	$A5	L	$CC	$8C	$CC	$8C
6&	$B6	$B6	$A6	$A6	M	$CD	$8D	$DD	$9D
7'	$B7	$B7	$A7	$A7	N^	$CE	$8E	$DE	$9E
8(	$B8	$B8	$A8	$A8	O	$CF	$8F	$CF	$8F
9)	$B9	$B9	$A9	$A9	P@	$DØ	$9Ø	$CØ	$8Ø
:*	$BA	$BA	$AA	$AA	Q	$D1	$91	$D1	$91
;+	$BB	$BB	$AB	$AB	R	$D2	$92	$D2	$92
,<	$AC	$AC	$BC	$BC	S	$D3	$93	$D3	$93
-=	$AD	$AD	$BD	$BD	T	$D4	$94	$D4	$94
.>	$AE	$AE	$BE	$BE	U	$D5	$95	$D5	$95
/?	$AF	$AF	$BF	$BF	V	$D6	$96	$D6	$96
A	$C1	$81	$C1	$81	W	$D7	$97	$D7	$97
B	$C2	$82	$C2	$82	X	$D8	$98	$D8	$98
C	$C3	$83	$C3	$83	Y	$D9	$99	$D9	$99
D	$C4	$84	$C4	$84	Z	$DA	$9A	$DA	$9A
E	$C5	$85	$C5	$85	→	$88	$88	$88	$88
F	$C6	$86	$C6	$86	←	$95	$95	$95	$95
					ESC	$9B	$9B	$9B	$9B

表B-2 標準的ASCII碼，Bit 8設定爲0

BIT NUMBERS								0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b_7 ↓	b_6 ↓	b_5 ↓	b_4 ↓	b_3 ↓	b_2 ↓	b_1 ↓	COLUMN → / ROW ↓	0	1	2	3	4	5	6	7
			0	0	0	0	0	NUL	DLE	SP	0	@	P		p
			0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
			0	0	1	0	2	STX	DC2	"	2	B	R	b	r
			0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
			0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
			0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
			0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
			0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
			1	0	0	0	8	BS	CAN	(	8	H	X	h	x
			1	0	0	1	9	HT	EM	)	9	I	Y	i	y
			1	0	1	0	A	LF	SUB	*	:	J	Z	j	z
			1	0	1	1	b	VT	ESC	+	;	K	[	k	{
			1	1	0	0	C	FF	FS	,	<	L	\	l	¦
			1	1	0	1	d	CR	GS	—	=	M	]	m	}
			1	1	1	0	E	SO	RS	.	>	N	^	n	~
			1	1	1	1	F	SI	US	/	?	O	—	o	DEL

附錄〔C〕 參考書目

R1：APPLE COMPUTER INC, APPLE Ⅱ REFERENCE MANUAL

R2：William E. Dougherty, THE APPLE Ⅱ MONITER PEELED（儒林）

R3：Steve Wozniak, SYSTEM DESCRIPTION：THE APPLE Ⅱ, MAY, 1977, BYTE

R4：Rockwell INC, 6500 HARDWARE MANUAL（宏碁）

R5：R. Zaks, PROGRAMMING THE 6502

R6：Allen Watsom Ⅲ, MORE COLORS FOR YOUR APPLE, JUNE, 1979, BYTE

R7：B.D. BLACKWOOD & G.H. BLACKWOOD, APPLESOFT LANGUAGE（儒林）

R8：POOLE, MCNIFF, COOK, APPLE Ⅱ USER'S GUIDE（儒林）

為研究開發發表新成果
為工廠引介最新科技實用資料
為學校教科書推陳出新

全華獻給您600餘種
科技圖書・
目錄備索・

全華科技圖書股份有限公司
台北市龍江路76巷20-2號2樓(台北總公司)
電話：581-1300・564-1819 郵撥：100836

1 電子學・電子電路

編號	書名	作者	定價
249	最新科技電子學(上)	江金柱	210
334	最新科技電子學(下)	江金柱	250
034	電晶體電路(上)、(下)	郭雲龍	各110
003	電子電路(上)、(下)	傅勝利	各170
319	電子電路總集(1)	黃孫金	210
366	電子電路總集(2)	黃孫金	230
133	實用電子電路(3)、(4)、(5)	張博堯	各160
352	實用電子電路(6)	張博堯	120
038	實用電晶體電路設計	陳連春	170
230	固態元件電路設計工程手冊	戴奎生	150
457	電子基本工作法	張政甫	190
247	電子設備故障檢修(上)、(下)	郭雲龍	各150
335	電子設備故障檢修(下)	郭雲龍	150
475	電子電路故障分析與檢修	蔡錦福	170

2 線性積體電路・音響・電視

編號	書名	作者	定價
005	積體電路學	孫康成	220
494	運算放大器手冊	歐福源等	150
236	運算放大器原理與應用(原理篇)	蔡中 魏哲和	100
314	運算放大器原理與應用(應用1篇)	張慧中 劉淙漢	90
148	線性IC實用手冊	徐茂林	平210 精250
272	IC定時電路設計手冊	李文麟等	120
053	IC型電源電路	簡章華	110
056	AM-FM收音機原理	黃政協	120
040	錄音機原理	黃政協	110
035	音響技術(上)、(下)	簡章華	各100
037	Hi-Fi擴大器	曾金龍	100
039	實用IC擴大器	徐兆達	120
235	音響系統	蔡明介	130
365	音響工程	許新添	110
289	彩色電視學	蔡澄雄	143
312	最新實用彩色電視學	葉嘉生	160
120	全晶體彩色電視檢修	黃克仁	110
151	超高頻電視調諧器	戴奎生	180
166	電腦選台電子調諧器	劉松生	120
344	錄放影機原理與檢修	黃國興	180
394	CTV數位選台技術	何時慧	220

3 數位電路・微電腦・計算機

編號	書名	作者	定價
291	組合邏輯	任建葳	169
330	序向邏輯	任建葳	150
242	數位電路與邏輯設計(上)、(下)	于憲厥	各160

383	基本硬體技術	張博堯	130
293	CMOS電路原理與應用	楊武智	140
267	CMOS電路基本實作(上)	張博堯	100
385	TTL IC特性手册(含互換表)	編輯部	280
307	CMOS電路故障分析與檢修	張博堯	130
201	數位儀器製作	張博堯	180
245	數位電路設計與硏究	張博堯	180
318	電子計算器淺說	林緒德等	120
216	電子錶原理設計與修護	邱岳輝	110
241	微電腦入門	陳范成	100
347	微電腦導論	侯瑞甫	110
380	微電腦基本原理	藍英杰	110
447	最新微電腦辭典	柯順隆	190
117	微計算機原理與應用	許瑞祥	220
477	圖解Z-80原理與使用	黃建聰	120
208	Z-80微算機原理與應用	楊世靖	110
167	微電腦系統與應用	劉飛生	230
497	Apple Ⅱ程式集—Game與應用	施威銘	140
504	8BIT微處理器實用系統硬體設計	李逸文	160
502	電腦語言—BASIC實用入門	張耀騰	140
355	計算機導論	陳憲昇	160
206	數位計算機電子學	薛福隆	190
465	中文電腦漫談	朱邦復	70

4 通信·電子儀表

編號	書名	作者	定價
118	無線電通信	劉春和	140
028	電訊系統	潘純新	160
041	示波器Ⅰ、Ⅱ	李台生	各110
066	電子儀表(上)、(下)	李台生等	各122
239	實用直流電源供應器	吳兆坤	120
508	轉換式電源供給器	簡章華	170
224	數位電子儀表	陳錫棋	140
459	電子儀器電路設計	陳錫棋	230
154	精密電子儀器	李順敦、徐永德等	平220 精240

5 工業電子·自動控制

編號	書名	作者	定價
020	工業電子學設計與應用	歐文雄	170
446	工業電子裝置與系統	李適中	240
320	閘流體實用手册	張繼石	210
046	實用SCR控制電路分析	徐茂林	90
329	家庭電器電子控制	林逸人	140
367	工業計器與伺服裝置	林逸人	140
233	工業機械電子控制	林逸人	140
368	數值控制與工作母機	林逸人	140
402	工業電子技能檢定	張政甫	320
506	微電腦電機順序控制實習	賴日生	180

⊙本目錄爲七十三年定價，爾後若有調價，請以最新目錄爲準。

全華圖書　版權所有　翻印必究
局版台業字第0223號　法律顧問：陳培豪律師

Apple II 程式集
— Game與應用

施威銘　編著

出版者　全華科技圖書股份有限公司
北市龍江路76巷20-2號
電話：581-1300・541-5342
581-1362・581-1347
郵撥帳號：100836
發行人　陳　本　源
印刷者　華一彩色印刷廠
定　價　新臺幣 140 元
再　版　中華民國73年3月

全華電腦相關叢書

363 **電腦概論與BASIC原理與應用**
丘祺中編著
16K／153頁／180元

445 **BASIC電腦程式語言**
葉得祥編著
16K／333頁／230元

458 **電腦語言 PASCAL 實用入門**
林同聖編譯
25K／225頁／120元

447 **最新微電腦辭典**
柯順隆編譯
32K／246頁／190元

465 **中文電腦漫談**
朱邦復編著
25K／100頁／70元

466 **零壹中文電腦程式語言**
零壹科技顧問公司編寫
25K／65頁／60元

483 **結構化程式語言——PASCAL**
葉垂奇編著
25K／336頁／180元

493 **Apple II Basic商用電腦程式**
朱力平•吳龍鎮編譯
25K／192頁／120元

●上列書籍爲七十三年定價，爾後若有調價，請以最新目錄爲準

圖書編號　023497

Apple Ⅱ程式集—Game與應用

施威銘 編著

全華圖書

497